Titre: Title:	Application d'un système probabiliste bayésien pour prédire la moyenne cumulative des étudiants à l'École Polytechnique de Montréal
Auteur: Author:	Antoine Murry
Date:	2016
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Murry, A. (2016). Application d'un système probabiliste bayésien pour prédire la moyenne cumulative des étudiants à l'École Polytechnique de Montréal [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/2210/

UNIVERSITÉ DE MONTRÉAL

APPLICATION D'UN SYSTÈME PROBABILISTE BAYÉSIEN POUR PRÉDIRE LA MOYENNE CUMULATIVE DES ÉTUDIANTS À L'ÉCOLE POLYTECHNIQUE DE MONTRÉAL

ANTOINE MURRY

DÉPARTEMENT DE GÉNIE ÉLECTRIQUE

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION

DU DIPLÔME DE MAITRISE ÈS SCIENCES APPLIQUÉES

(GÉNIE ÉLECTRIQUE)

JUIN 2016

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

APPLICATION D'UN SYSTÈME PROBABILISTE BAYÉSIEN POUR PRÉDIRE LA MOYENNE CUMULATIVE DES ÉTUDIANTS À L'ÉCOLE POLYTECHNIQUE DE MONTRÉAL

présenté par : MURRY Antoine

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. DESMARAIS Michel, Ph. D., président

M. BRAULT Jean-Jules, Ph. D., membre et directeur de recherche

M. BASSETTO Samuel-Jean, Ph. D., membre et codirecteur de recherche

M. BOUDRAULT Yves, Ph. D, membre

DÉDICACE

A ma famille pour son soutien durant la totalité de mes études,

En particulier mes parents Pierre et Edith,

Ainsi que ma sœur Elsa

REMERCIEMENTS

Je tiens à remercier mon directeur de recherche, Monsieur Jean-Jules Brault, pour son soutien lors du déroulement de ce projet.

Je tiens également à remercier mon co-directeur Monsieur Samuel Bassetto, pour avoir proposé le projet, et pour avoir effectué les démarches pour obtenir les données.

RÉSUMÉ

Dans un contexte où la puissance de calcul des ordinateurs est en constante augmentation, et où de plus en plus de données sont recueillies par les organisations, ces dernières s'intéressent de manière grandissante à l'exploration de leurs bases de données (Data Mining), dans le but d'améliorer leurs processus.
Par processus, nous entendons tout ensemble d'activités transformant un élément d'entrée en un élément de sortie.

Si la grande majorité des processus sont monitorés, et que des observations surviennent durant leur déroulement pour assurer leur contrôle, ces observations sont néanmoins rarement utilisées pour prédire l'état final. Ce travail s'intéresse ainsi à la prédiction de l'état de sortie des processus à partir des observations survenues durant leur déroulement.
Nous nous intéressons en particulier à la formation des étudiants au baccalauréat de l'École Polytechnique de Montréal. Nous considérons cette formation comme un processus, dans le sens où il s'agit d'un enchaînement d'étapes, sur 12 sessions (4 ans), transformant les étudiants entrants en des ingénieurs prêts à entrer sur le marché de l'emploi.
Les données disponibles sont, pour chaque étudiant, les notes moyennes obtenues à chacune des 12 sessions du baccalauréat, les crédits pris à chaque session, et le département de l'étudiant en question.

L'idée est de tenter de prédire la moyenne cumulative des six dernières sessions obtenue par chaque étudiant, à partir de ses moyennes obtenues à chacune des six premières sessions, du nombre de crédits pris durant chaque session, et du département de l'étudiant.

Ce travail s'intéresse ainsi à tester si les informations disponibles et mesurables sont suffisantes pour fournir une prédiction informative, en dépit d'informations non disponibles, telles que des variables reflétant la psychologie, la vie personnelle ou associative de l'étudiant. En particulier, nous nous intéressons à savoir s'il existe des formes (patterns) dans l'évolution des notes moyennes des étudiants au cours de leur baccalauréat.

Nous avons pour cela développé un système basé sur les réseaux bayésiens, qui sont des modèles probabilistes graphiques, permettant ainsi d'estimer la distribution des prédictions possibles en fonction du pattern particulier d'un étudiant. Le système peut alors servir d'outil d'aide à la décision.

Nous avons pu obtenir des données provenant de la cohorte de 2008, comportant 700 étudiants.
Deux types d'expériences ont été réalisées :

- Une première avec des données simulées, où nous avons généré des formes, c'est-à-dire des tendances, ou trajectoires, dans les évolutions des moyennes des étudiants au cours des sessions (par exemples tendances montantes, descendantes…). Le but étant de s'assurer du bon fonctionnement de l'approche (et calibrer les divers paramètres) avec des données contrôlées, mais aussi pour comprendre ses limites.

- Une seconde avec les données réelles, de manière à déterminer si des prédictions probantes peuvent effectivement être effectuées dans le contexte spécifique de l'École Polytechnique.

Dans les deux types d'expérience, nous avons utilisé la mesure « Log-Loss », qui évalue l'inexactitude de notre estimation probabiliste (distribution de la sortie fournie par le système) pour chaque étudiant testé.

Les résultats obtenus en données simulées montrent que le système parvient très bien à reconnaître les formes, mais a cependant une tendance à fournir des estimations probabilistes distordues, favorisant les valeurs les plus probables et négligeant les autres (comportement souvent observé dans les réseaux bayésiens)

L'expérience en données réelles suggère que, avec les données disponibles mesurables, le fait de combiner les moyennes obtenues aux six premières sessions n'est pas plus informatif que le fait de prendre en compte seulement une session pour prédire la moyenne obtenue lors de la seconde partie du baccalauréat. Nous pouvons également voir que plus la session est

avancée, plus la moyenne obtenue à cette session est prédictive, ce qui suggère que les comportements ont globalement tendance à se figer au fur et à mesure que le processus de formation avance.

Il serait intéressant de recueillir davantage de données de Polytechnique, à la fois en termes d'exemples, mais aussi en termes de variables d'entrées (notes pour chaque cours et non seulement la session) de manière à tester si une amélioration dans les performances survient.

ABSTRACT

As computing power is getting more and more important, and as more and more data is being gathered by organizations, the latter are gaining interest in mining their databases, in order to improve their processes.

By process is meant a set of activities which transform an input into an output.

If most processes are monitored, and if observations are gathered during process runs in order to keep them under control, those same observations are rarely ever used in order to predict the final state of processes. This research hence focuses on the prediction of the output of processes, based on the observations gathered during process runs.

We will bear a special interest for the education of bachelor students at Ecole Polytechnique de Montréal. We will consider it is a process, to the extent that it is a succession of 12 quarters (4 years), which transforms incoming students into engineers ready to start their careers.

The available data provides, for every single student, the grade point average (GPA) obtained at each quarter, the number of credits that were taken, as well as the Department the student belong to.

The idea is to try to predict the cumulative GPA obtained during the six last quarters, given the grades and the number of credits taken during the first six quarters, as well as the Department the student belongs to.

Hence, this research focuses on testing whether the available and measurable information is enough to provide an informative prediction, in spite of non-available information, such as variables reflecting the psychology, or the personal lives of all students. We will be particularly interested in discovering whether patterns are present in the trajectories of student GPAs, during the evolution of their bachelor.

We have developed a system based on Bayesian networks, which are probabilistic graphical models, hence allowing a probabilistic estimation of the predicted cGPA. The system can therefore be used as a decision aid.

We gathered data from the 2008 cohort, which consists of 700 students.

Two experiments were led:

- The first one, based on simulated data, where we have generated patterns, i.e. trends, or trajectories of student GPAs during their bachelors (for example rising or falling trends). The main purpose of this test is to make sure that the system works as expected, and to calibrate all parameters with controlled data, but also to highlight the limits of the proposed system.
- A second experiment with real data, in order to find out whether relevant predictions can indeed be obtained, in the specific case of Ecole Polytechnique.

In both cases, we used the « Log Loss » measure, which evaluates the error of the probabilistic estimation (probabilistic distribution of the output of the process) for each tested student.

The results obtained with simulated data show that the system does manage to recognize patterns, but, however, tends to provide overconfident probabilistic estimations (which is often the case with Bayesian networks).

The results obtained with real data suggest that, with the available and measurable data, combining the GPAs obtained during the first six quarters the bachelor does not provide predictions more informative than simply taking the GPA obtained during one of the first six quarters of the bachelor. We also observe that the more advanced the quarter, the more the obtained GPA is influent on the cGPA obtained during the second part of the bachelor. This suggests that the student behaviors tend to freeze as the bachelor process evolves.

It would be interesting to gather more data from Polytechnique, both in terms of examples, but also in terms of entrance variables (e.g. grade obtained at each course), in order to test whether a performance improvement occurs.

TABLE DES MATIERES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

La liste des sigles et abréviations présente, dans l'ordre alphabétique, les sigles et abréviations utilisés dans le mémoire ou la thèse ainsi que leur signification. En voici quelques exemples :

ANB	Augmented Naïve Bayes (Réseau Bayésien Naif Augmenté)
cGPA	Cumulative Grade Point Average (Moyenne Cumulative)
Dkl	Divergence de Kullback-Leibler
EDM	Educational Data Mining (Exploration de données pour l'éducation)
GPA	Grade Point Average (Moyenne par Session)
LL	Log-Loss
LLmoy	Log-Loss moyen
NB	Naïve Bayes (Réseau Bayésien Naif)
P (A)	Distribution de probabilités a priori de la variable aléatoire A
P (A | B)	Distribution de probabilités a posteriori de A sachant B
A ^ B	Conjonction de la variable aléatoire avec la variable aléatoire (« A ET B »)

LISTE DES ANNEXES

CHAPITRE 1 INTRODUCTION

Durant les cinquante dernières années, les organisations ont investi dans des programmes d'amélioration de la qualité. Plusieurs méthodes ont été développées et adoptées, comme la Maitrise Statistique des Procédés, la méthodologie Zéro Défauts, le Lean Six Sigma. Ces méthodes s'appliquent aux industries manufacturières, où la qualité est associée à l'usage final du produit, ainsi qu'aux industries de services, où la qualité se définit comme la satisfaction du client pour le service reçu.

Dans le but d'améliorer la qualité encore davantage, des méthodes basées sur la fouille des données industrielles se sont développées. Ces méthodes cherchent à identifier ou reconnaitre des formes dans les bases de données des processus, de manière à obtenir de l'information pour les améliorer davantage.

L'une de ces méthodes s'intéresse à la prédiction de la qualité. Cela signifie prédire la valeur d'un indicateur relatif à la qualité finale (état) d'un produit ou d'un service, à partir d'indicateurs reflétant l'état du processus durant son déroulement.

Avoir une prédiction de la qualité finale permet d'améliorer le processus en permettant une intervention préventive sur le processus, dans les cas où des défauts ont été prévus (Bouslah, Ghrabi, & Pellerin, 2014).

Par exemple, dans les industries des semi-conducteurs, où les processus peuvent prendre plusieurs semaines avant de finir, une perte de quelque pourcents sur la production implique des pertes financières s'élevant à des millions de dollars (Weiss, Dhurandhar, & Baseman, 2013). Ainsi, ces industries ont considérablement investi dans des systèmes de prédiction des défauts des puces en production.

Dans l'industrie des services, il est possible de trouver des exemples suivant un schéma similaire. Ainsi, la formation d'étudiants durant leur baccalauréat suit un processus long (4 ans), et les clients (les étudiants) peuvent dépenser une somme considérable pour financer leurs cours. Certaines universités s'intéressent à la prédiction des succès ou difficultés des étudiants lors de leurs études, de manière à lutter plus efficacement contre l'échec scolaire (Romero & Ventura, 2010).

1.1 Problématique

Plusieurs études se sont intéressées à la mise en place d'outils pour améliorer la qualité de sortie des processus.

Par processus, nous entendons tout ensemble d'activités transformant un élément d'entrée en un élément de sortie.

Par exemple, dans le domaine manufacturier, le modèle de (Bouslah, Ghrabi, & Pellerin, 2014) se base sur une estimation des taux de panne pour optimiser la taille du lot qui va minimiser les pertes. Le travail de (Baud-Lavigne, Bassetto, & Agard, 2014) base les fréquences d'échantillonnage sur les évolutions du nombre de rebuts à la sortie des processus de production. Le travail de (Bassetto, Paredes, & Baud-Lavigne, 2013) a étendu l'étude précédente en liant le taux d'apprentissage au nombre de contrôles internes durant le processus. Ce lien permet une modélisation plus fine de la quantité optimale de contrôles nécessaires concernant le rendement en sortie du processus. Cela permet donc une simulation plus fine de l'évolution de la performance d'un processus de production, en fonction des observations qui sont menées durant ce processus. Cependant, les données mesurées durant le processus ne sont pas employées pour prédire la qualité en sortie, et éventuellement donner la quantité de contrôles nécessaires.

Les travaux de (Sahnoun, Bettayeb, & Bassetto, 2014) présentent également une optimisation du plan d'inspection pour minimiser la quantité de contrôles en garantissant un seuil de risque en sortie de sortie le plus bas possible. Cependant, là aussi, les données de ces contrôles ne sont pas associées à la prédiction de la performance en sortie du processus. Seul l'ordonnancement des contrôles est utilisé pour définir son importance.

Les travaux de (Bettayeb, Bassetto, Vialletelle, & Tollenaere, 2012) s'inscrivent également dans cette perspective : réordonnancer et réallouer les capacités de contrôle qualité durant le processus afin de minimiser le risque en sortie du processus. Cependant, aucun usage n'est fait de ces données pour prédire réellement la qualité en sortie.

Enfin, une autre classe de travaux (Restrepo-Moreno, Charron-Latour, Pourmonet, & Bassetto, Accepted 2015) s'intéresse à la saisie des opportunités d'amélioration, menées pas le personnel, pour localiser les actions d'amélioration sur le processus de production et contribuer à une meilleure performance. Cependant, cette méthode n'évalue pas la qualité en sortie.

En regard de ces travaux, nous nous sommes penchés sur comment utiliser l'information obtenue par ces contrôles qualité et mesures intermédiaires pour prédire la qualité en sortie et aussi intervenir dans la prise de décision vis-à-vis du processus de production.

Prédire l'avenir à partir d'observations disponibles est une tâche qui peut se révéler extrêmement incertaine. En effet, les indicateurs disponibles reflétant l'état du processus durant son évolution ne sont généralement pas les seuls éléments pouvant expliquer l'évolution future du processus. Cela est d'autant plus vrai dans le cas des processus impliquant des comportements humains. La prise de décision devrait ainsi prendre en compte l'incertitude de la prédiction. L'utilisation de méthodes probabilistes permet de fournir des prédictions reflétant l'incertitude (Bessière, Mazer, Ahuactzin, & Mekhnacha, 2013)

Ainsi, nous nous sommes posé la problématique suivante : comment obtenir une prédiction probabiliste sur l'état de sortie d'un processus en cours, sachant les valeurs des variables observées durant son déroulement ?

De manière à proposer une réponse à cette question de recherche, nous nous sommes intéressés au développement d'un système de prédiction probabiliste basé sur les réseaux bayésiens.

Les réseaux bayésiens sont des modèles probabilistes graphiques. Ils sont reconnus pour pouvoir être compréhensibles par les experts des domaines d'application, de par leur aspect graphique.
De plus, leur aspect probabiliste leur permet de se présenter comme une méthode adaptée pour retransmettre l'incertitude lors des prédictions.

Ce travail s'intéresse au cas du processus de formation des étudiants lors du baccalauréat à l'Ecole polytechnique de Montréal. L'idée est de prédire, pour chaque étudiant, la moyenne cumulative obtenue lors de la seconde partie du baccalauréat, sachant les notes moyennes obtenues, le nombre de crédits pris lors de la première partie de la formation (6 premières sessions), ainsi que le département.

1.2 Objectifs de recherche

Ce projet s'articule autour des objectifs suivants :

1) Développer un outil intégrant une implémentation de réseaux bayésiens avec une base de données et une sortie graphique visualisant la prédiction probabiliste, permettant ainsi une aide à la décision.

2) Tester la capacité de l'outil à reconnaitre des formes et à ressortir une estimation probabiliste de l'état de sortie (moyenne cumulative) avec le processus de formation des étudiants au baccalauréat, à partir de données simulées. Cela permettra de vérifier le fonctionnement du système et de comprendre ses limites.

3) Appliquer le système sur des données réelles, de manière à tester s'il détecte des tendances et s'il parvient effectivement à prédire la sortie de manière probabiliste, pour le cas du baccalauréat à Polytechnique Montréal.

1.3 Cas d'application et Méthode

Cas d'application :

De manière à pouvoir répondre aux objectifs 2) et 3), l'école nous a transmis les données contenant les notes de tous les étudiants de la cohorte de 2008 des étudiants au baccalauréat. Nous avons accès, pour chaque étudiant, à sa moyenne par session, au nombre de crédits pris à cette session, ainsi qu'au département auquel il appartient.

L'objectif de ce cas d'application est d'utiliser le système proposé pour tenter d'obtenir une prédiction probabiliste sur la moyenne cumulée obtenue par un étudiant donné durant la seconde partie de son baccalauréat (session 7 à session 12) qui est associée à la qualité finale du processus, sachant les notes obtenues lors de la première moitié du baccalauréat (session 1 à la session 6), le nombre de crédits pris à chacune des sessions passées et le département de l'étudiant.

Il s'agit d'un cas qui s'annonce particulièrement complexe, en ce qui concerne l'incertitude liée aux prédictions, car il nous manque de l'information (relative par exemple à la vie personnelle des étudiants) pour pouvoir prédire précisément la qualité finale, d'où le besoin d'un système probabiliste.

L'hypothèse de fonctionnement de ce cas d'application est que les moyennes par session des étudiants au cours de leur baccalauréat forment des trajectoires, auxquelles correspond une valeur de qualité finale (une moyenne cumulée finale) associée.

- Les résultats obtenus avec les données simulées montrent que le système parvient bien à reconnaitre des formes. Cependant, nous mettons en évidence que l'estimation probabiliste de la qualité de sortie est biaisée, dans le sens où les valeurs les plus probables sont favorisées par rapport à la réalité de la proportion.
- Avec les données réelles, nous montrons que le système ne parvient pas, en fusionnant l'information provenant des six premières sessions, à fournir des estimations probabilistes de la sortie globalement plus informatives que celles fournies par la simple distribution a posteriori de la sortie sachant la valeur de la dernière observation (moyenne obtenue à la session 6).

Indicateur :

De manière à tester la performance de la proposition nous utiliserons la mesure appelée « log-loss ». Cette dernière évalue la justesse de la prédiction probabiliste fournie, plutôt que le nombre ou le pourcentage de classifications correctes.

1.4 Structure du Mémoire

La suite du mémoire se divise en cinq parties :

- Le deuxième chapitre est la revue de littérature. Nous nous intéresserons aux différentes pratiques mises en œuvre pour la prédiction de l'état de sortie des processus, dans diverses industries et organisations. Nous présenterons également les réseaux bayésiens, leurs points d'intérêts, ainsi que leurs domaines d'applications principaux.
- Le troisième chapitre présente le système proposé, c'est à dire la manière dont il s'intègre dans un processus, ainsi que a structure.
- Le quatrième chapitre présente le cas d'application, du processus de formation de Polytechnique Montréal, en détail.
- Le cinquième chapitre s'intéresse à la démarche suivie pour tester l'habilité du système à détecter les formes et à fournit une estimation probabiliste de la sortie. Nous détaillerons la génération des données simulées, et la méthode suivie pour l'application sur les données réelles. Nous analyserons enfin les résultats obtenus lors des tests.
- Le sixième et dernier chapitre est la conclusion, qui proposera les axes pour la recherche future.

CHAPITRE 2 REVUE DE LITTÉRATURE

La revue de littérature présente dans un premier temps le contexte de la prédiction de la qualité dans la fouille de données industrielle. Elle présente ensuite les Réseaux Bayésiens, les raisons pour lesquelles ils sont utilisés en pratique, leurs avantages et inconvénients, ainsi que les principaux domaines dans lesquels ils sont appliqués. Enfin, le chapitre présentera les différentes méthodes utilisées pour la prédiction de la qualité dans l'industrie.

2.1 L'exploration de données pour l'analyse prédictive

L'analyse prédictive peut se présenter comme un cas particulier de l'exploration de données ou Data Mining. Cette dernière est-elle même une étape du processus de découverte de connaissances dans les bases de données (Knowledge Discovery in Databases, ou KDD).

Le processus KDD a pour objectif de fouiller des bases de données industrielles, de manière à faire ressortir de l'information permettant une aide à la décision (Fayyad, Piatetsky-Shapiro, & Smyth , 1996).

Il se décompose habituellement en cinq étapes, décrites dans la figure 2.1 (Fayyad, Piatetsky-Shapiro, & Smyth , 1996).

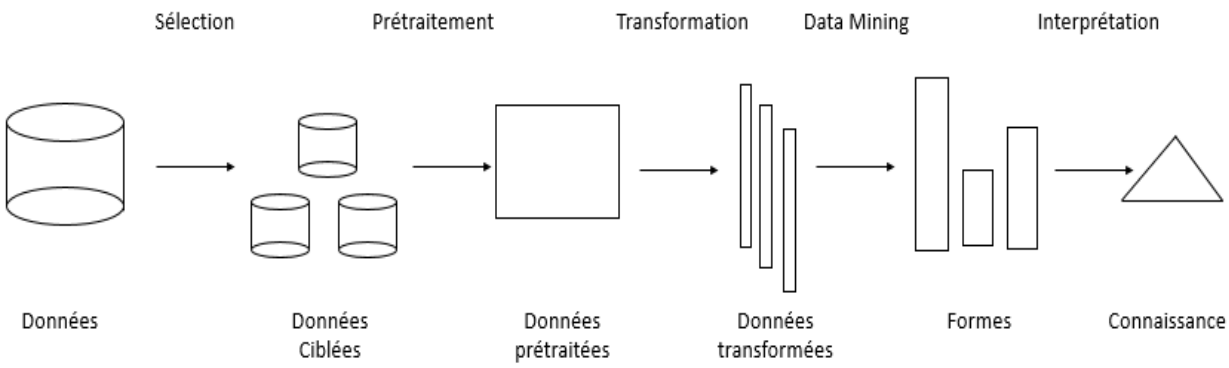


Figure 2-1: les 5 étapes du Processus KDD (tiré de http://www2.cs.uregina.ca/~dbd/cs831/notes/kdd/1_kdd.html)

La première étape est relative à la sélection des données à prédire, en fonction des besoins industriels, et des informations disponibles dans la base de données.

La seconde étape est relative au prétraitement des données, c'est-à-dire leur nettoyage des bases de données (par exemple supprimer les valeurs aberrantes, remplacer les valeurs manquantes).

La troisième étape consiste à transformer les données en vue de leur utilisation par des algorithmes dans l'étape suivantes. On s'intéresse souvent à projeter les données vers des dimensions plus informatives, ou à normaliser les données pour faciliter l'apprentissage machine.

La quatrième étape est l'étape de la fouille de données. C'est l'étape où sont appliqués les algorithmes pour découvrir ou reconnaitre des formes dans la base de données. La fouille de données se décompose généralement en six taches (Fayyad, Piatetsky-Shapiro, & Smyth , 1996):

- La Classification, qui consiste à prédire la classe d'une variable d'intérêt, quand cette dernière est catégorique.
- La Régression, qui consiste à prédire la valeur numérique d'une variable d'intérêt, quand cette dernière est réelle.
- La détection d'Anomalies, qui consiste à détecter des erreurs ou des anomalies.
- La découverte de règles d'associations, qui consiste à chercher à établir des relations entre des variables, de manière à comprendre davantage le fonctionnement des systèmes étudiés.
- La segmentation, qui consiste à regrouper des instances ayant des propriétés similaires.
- La visualisation, qui consiste à représenter de manière synthétique des informations découvertes dans des bases de données.

La dernière étape consiste à interpréter et à transmettre les informations découvertes, en vue d'une aide à la décision.

Ce projet s'intéresse à la prédiction de la qualité finale des processus. Ainsi, ceci concerne deux des six tâches de fouille de donnée citées : la Classification, dans le cas où les indicateurs de l'état de sortie sont catégoriques (e.g. Présence de Défaut ou Non), et la Régression, dans le cas où l'indicateur est numérique.

La Régression et la Classifications sont basées sur l'utilisation d'algorithmes appelés algorithmes d'apprentissage supervisé.

Dans le cadre de ce travail, nous allons nous intéresser aux processus suivant le schéma décrit dans la Figure 2.2.

Ce schéma est inspiré de l'implantation des systèmes de prédiction de qualité dans les processus manufacturiers.

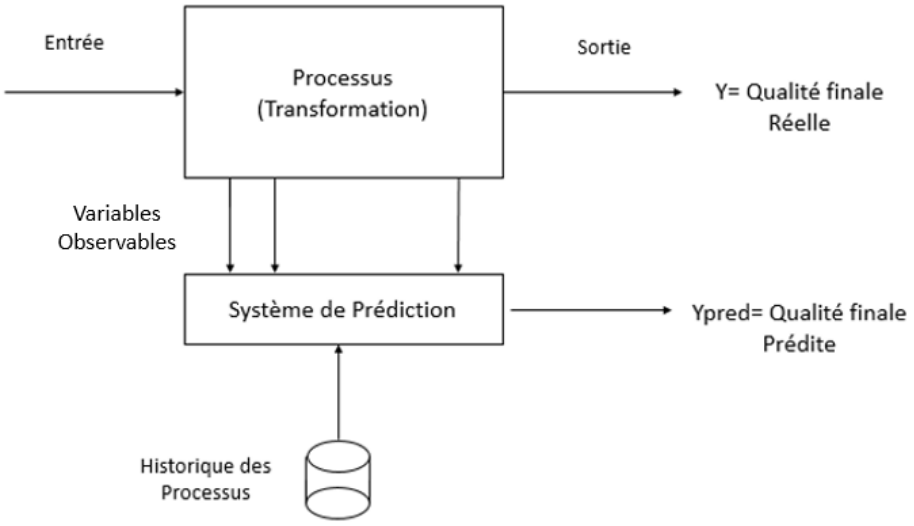


Figure 2-2 : Système de Prédiction. Inspiré de (Khan, Moyne, & Tilbury, An Approach for Factory-Wide Control Utilizing Virtual Metrology, 2007)

Les processus, par définition, transforment une entrée en une sortie. La qualité de la sortie se définit par un indicateur numérique reflétant l'état de la sortie. Dans le cas d'un produit, cela correspond le plus souvent au bon fonctionnement du produit, selon les normes en vigueur. Dans le cas d'un service, cela correspond à la satisfaction du client.

D'autre part, les processus sont souvent suivis par des indicateurs, qui sont relevés durant leur déroulement, selon une fréquence d'échantillonnage définie. Dans ce mémoire, nous appellerons ces indicateurs les variables observables.

De manière à pouvoir appliquer des méthodes de prédiction par apprentissage statistique, un historique des processus passés est requis. Nous avons besoin d'une base de données comprenant pour chaque transformation de l'historique, les valeurs des différentes variables observables, ainsi que la valeur de la qualité finale correspondante.

L'apprentissage supervisé consiste ici à appliquer un algorithme pour tenter d'apprendre la relation entre la qualité de sortie du processus et les variables de processus. Ainsi, si l'apprentissage est

efficace, il est alors possible d'obtenir une prédiction en temps réel sur la sortie avant la fin du processus, et donc d'intervenir de manière préventive dans les cas où des défauts sont prévus.

Comme nous l'avons mentionné en introduction, obtenir une prédiction juste peut se révéler extrêmement complexe, en grande partie à cause de l'incertitude, liée au fait que les variables observables seules ne suffisent souvent pas à expliquer la qualité finale.

Différents modèles, basées sur différents algorithmes peuvent être employés pour obtenir une prédiction. Parmi ces modèles figurent les réseaux bayésiens.

Un point important doit être mentionné. La problématique qui nous intéresse se focalise sur les bases de données contenant un nombre important d'exemples, et un nombre plus faible de variables d'observations. Ainsi, nous ne nous intéresserons pas aux séries temporelles qui elles, se focalisent sur des processus ayant un nombre restreint d'exemples, mais un nombre très important d'observations, et qui tentent de détecter des répétitions ou des cycles dans ces grandes séries d'observations.

Nous allons maintenant présenter les réseaux bayésiens, leurs domaines d'applications principaux, ainsi que leurs forces et faiblesses.

2.2 Présentation des Réseaux bayésiens, et leurs domaines d'utilisation

2.2.1 Présentation des Réseaux Bayésiens, de leurs avantages et inconvénients

Les réseaux bayésiens sont des modèles probabilistes graphiques. Leur format graphique est composé de nœuds et d'arcs directionnels. Les nœuds représentent des variables aléatoires Xi, et les arcs les corrélations entre ces variables aléatoires. Il s'agit de modèles acycliques, c'est-à-dire qu'il ne peut pas y avoir de chemin tels que Xi -> … -> Xn, tels que Xi=Xn. (Bishop, 2006).

Plus précisément, un réseau bayésien est une représentation graphique de la distribution conjointe P(X1, …, Xn). Une distribution conjointe permet de modéliser les interactions entre toutes les

variables. En pratique, on cherche à décomposer la distribution conjointe, de manière à ne pas prendre en compte les interactions inexistantes.

La représentation graphique permet de proposer un langage facilitant la compréhension de la décomposition, et donc des relations entre les variables. Il s'agit de l'aspect qualitatif du modèle.

D'autre part, chaque nœud comporte une table de probabilités associée, qui définit sa relation avec les nœuds dits « parents », c'est-à-dire dont les arcs sont pointés sur la variable en question. Il s'agit de l'aspect quantitatif du modèle.

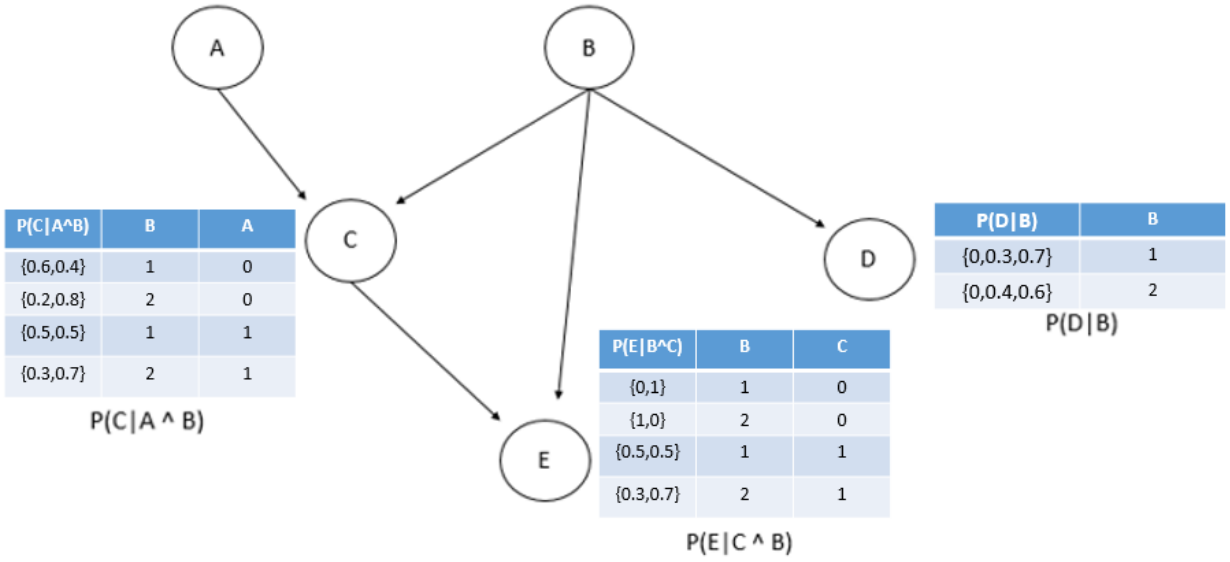


Figure 2-3: Exemple de Réseau Bayésien

La figure 2.3 montre un exemple de Réseau Bayésien. Les nœuds A, B, C, D et E représentent les variables aléatoires. Nous avons également représenté les tables de probabilité associées. Par exemple, la table associée au nœud P(C|A^B) présente les probabilités a posteriori dans la colonne de gauche (C prend deux valeurs, donc une probabilité est associée à chaque valeur). La colonne du milieu (B, qui prend deux valeurs : 1 et 2) et la colonne de droite (A, qui prend deux valeurs : 0 et 1) présentent les valeurs des variables parents.

Ce réseau bayésien est la représentation graphique de la distribution décomposée de la façon suivante (notation reprise de (Bessière, Mazer, Ahuactzin, & Mekhnacha, 2013)) :

$$P(A \wedge B \wedge C \wedge D \wedge E) = P(A) \times P(B) \times P(C|A \wedge B) \times P(E|C \wedge B) \times P(D|B)$$

Ainsi, chaque terme de la distribution conjointe décomposée correspond à une table de probabilités.

Le but des réseaux bayésiens est de modéliser et de quantifier les relations d'un phénomène ou d'un système (par exemple un processus), de manière à pouvoir inférer, de manière probabiliste, des valeurs sur des variables d'intérêt, sachant des valeurs de variables que l'on a déjà sur le phénomène en question.

Cette capacité à inférer des valeurs de variables d'intérêt est rendue possible grâce à une méthode appelée l'inférence bayésienne, qui permet de calculer les distributions de probabilités a posteriori, sachant les valeurs des variables connues.

Par exemple, si l'on souhaite connaitre la distribution a posteriori $P(E \mid A \wedge B \wedge C \wedge D)$, l'inférence bayésienne consiste à effectuer l'opération suivante, basée sur la formule de Bayes (Bessière, Mazer, Ahuactzin, & Mekhnacha, 2013):

$$P(E \mid A \wedge B \wedge C \wedge D) = \frac{P(A \wedge B \wedge C \wedge D \wedge E)}{\sum_E P(A \wedge B \wedge C \wedge D \wedge E)} = \frac{P(A) \times P(B) \times P(C \mid A \wedge B) \times P(E \mid C \wedge B) \times P(D \mid B)}{\sum_E P(A) \times P(B) \times P(C \mid A \wedge B) \times P(E \mid C \wedge B) \times P(D \mid B)}$$

Nous voyons que le calcul est principalement basé sur l'intégration de la distribution conjointe. Si cette dernière est connue, n'importe quelle variable du réseau peut être inférée.

Le graphique des relations entre les variables peut être déterminé automatiquement, ou bien par les experts des systèmes en question. Cette dernière méthode permet à ces experts de transmettre leurs connaissances au modèle. Cela a pour bénéfice d'une part de simplifier les calculs d'inférence bayésienne, en supprimant manuellement les corrélations entre les variables indépendantes, et d'autre part, d'obtenir des modèles qui peuvent être compris par leurs utilisateurs, et donc davantage acceptés que des modèles produits par des méthodes « boîtes noires », comme les Réseaux de Neurones (Lee & Abbott, 2003).

L'aspect quantitatif, quand à lui, est le plus souvent appris automatiquement, à partir des bases de données disponibles, car il est souvent long ou impossible de transmettre correctement cette information au modèle manuellement.

Si l'aspect graphique des réseaux bayésien permet de proposer des modèles compris et acceptés par les experts du système en question, leur aspect probabiliste permet de proposer une méthode adaptée pour raisonner dans des situations comportant de l'incertitude (Bessière, Mazer, Ahuactzin, & Mekhnacha, 2013).

En effet, le fait de présenter les prédictions sous forme de distributions de probabilités a posteriori permet de représenter l'incertitude sur la prédiction.

De manière à illustrer ce point, deux exemples extrêmes peuvent être cités. Une prédiction certaine conduira à une distribution de probabilités de type Dirac, centrée sur la valeur prédite pour la variable d'intérêt. En revanche, si l'incertitude est maximale, le modèle retournera une distribution uniforme sur l'ensemble des valeurs possibles pour la variable à prédire. En pratique nous obtiendrons des distributions de probabilité se situant entre ces deux extrêmes.

La représentation de la prédiction sous forme de distribution permet ainsi une prise de décision tenant compte des probabilités de chaque valeur de la variable à prédire. Par exemple, une prédiction binaire représentant la qualité finale d'un produit en cours de production ayant une probabilité de 0.99 sur la valeur « 1 » (: qualité acceptable) et 0.01 sur la valeur « 0 » (: rebut) indique une situation où l'on peut être certain que la qualité finale sera acceptable. En revanche, des valeurs de probabilités pour « 0 » et « 1 » de respectivement 0.45 et 0.55 indique une situation incertaine, dans le sens où un risque que la qualité de sortie soit mauvaise est présent, d'où un besoin d'éventuellement intervenir sur le processus en cours, même si la probabilité maximale correspond à la valeur « 1 ».

Outre l'aspect graphique permettant la participation d'experts, et l'aspect probabiliste, permettant la prise en compte de l'incertitude, les réseaux bayésiens présentent d'autres avantages, ainsi que des inconvénients.

Un autre avantage notable des réseaux bayésiens est qu'ils ne sont pas sensibles aux problèmes « mal posés », notamment en ce qui concerne les problèmes dits « inverses » (Bessière, Mazer, Ahuactzin, & Mekhnacha, 2013). Cela signifie qu'un problème comportant plusieurs solutions, qui est insoluble avec de nombreuses méthodes d'apprentissage statistique (selon le concept du problème bien posé), pourra être abordé par un réseau bayésien, qui présentera les différentes solutions avec leurs probabilités associées.

Ainsi, un réseau bayésien sera apte à inférer des valeurs pour des problèmes dits « directs », dont la solution est unique, mais également les problèmes « inverses », comportant plusieurs solutions.

En ce qui concerne les désavantages, l'un des problèmes majeurs est, dans le cas de phénomènes complexes, la complexité de modélisation par les experts, qui peuvent faire des erreurs (Morgan & Henrion, 1990).

Aussi, dans le cas de réseau complexes, l'inférence bayésienne peut se révéler très couteuses en termes de temps de calcul, voire, impossible. Il faut alors recourir à des méthodes d'inférence approximative (Bessière, Mazer, Ahuactzin, & Mekhnacha, 2013).

Enfin, comme nous le verrons par la suite, les réseaux bayésiens peuvent faire face à des difficultés lors de la présence de variables continues (notamment pour déterminer les tables de probabilités des variables ayant une variable parent continue). Celles-ci peuvent être discrétisées, ce qui peut mener à des difficultés si cette étape est mal effectuée. Fenton (Fenton, Neil, & Marquez, 2008) considère ce point comme le principal point faible des réseaux bayésiens.

Le tableau 2.1 liste les principaux avantages et inconvénients des réseaux bayésiens pour la modélisation.

Table 2-1: Avantages et Faiblesses des réseaux bayésiens pour la modélisation

Avantages	Inconvénients
- Aspect Graphique, permettant au modèle d'être compris par ses utilisateurs (Heckermann, Geiger, & Chickering, 1995)	- La création manuelle du modèle peut prendre du temps aux experts, et ceux-ci peuvent se tromper (Morgan & Henrion, 1990)
- Aspect Probabiliste, permettant de travailler efficacement avec des situations incertaines (Heckermann, Geiger, & Chickering, 1995)	- L'inférence bayésienne peut se révéler très complexe voire impossible dans certain cas (Bessière, Mazer, Ahuactzin, & Mekhnacha, 2013). Besoin d'approximer.
- La création « manuelle » du modèle permet de travailler avec des bases de données relativement restreintes, car pas besoin d'apprendre la structure (Eisenstein & Alemi, 1996)	- Les variables continues doivent être discrétisées (Friedman & Goldszmidt, 1996)
- Robustesse face aux données manquantes (Kontaten et al, 1997)	
- Permettent d'inférer les valeurs de n'importe quelle variable du réseau (donc pas seulement de la prédiction) (Bessière, Mazer, Ahuactzin, & Mekhnacha, 2013)	
- Aspect Probabiliste permet la robustesse face aux problèmes mal posés (Bessière, Mazer, Ahuactzin, & Mekhnacha, 2013)	

2.2.2 Applications des Réseaux bayésiens

Les réseaux bayésiens sont appliqués dans différents domaines.

L'un des domaines principaux d'application est le domaine de la santé (Lee & Abbott, 2003). Ils sont par exemple populaires pour leur habilité à transmettre des prédictions pour des diagnostics, de manière probabiliste (Lucas, Van Der Gaag, & Hanna, 2004). Ils sont aussi utilisés en bio-modélisation, pour représenter les interactions entre les gènes.

Un autre domaine est la robotique. Les réseaux bayésiens sont utilisés pour créer des modèles pour les robots, de manière à ce qu'ils puissent apprendre à se déplacer dans leur environnement, en dépit d'incertitudes, comme des obstacles imprévus (Lebeltel, Bessière, Diard, & Mazer, 2004). L'une des méthodes les plus utilisées dans le domaine est la fusion d'information, qui consiste à intégrer différentes sources d'informations, provenant de sources souvent peu fiables (par exemple des capteurs endommagés), de manière à obtenir une prédiction probabiliste globale plus fiable, pour permettre de meilleures prises de décision. La fusion d'information est la méthode que nous utiliserons pour notre système.

Les réseaux bayésiens sont également utilisés dans l'industrie manufacturière, pour modéliser les risques, et prédire les défauts sur les équipements via un indicateur de santé (Bouaziz, Zamai, & Duvivier, 2013).

Dans le domaine de la qualité, les réseaux bayésiens sont globalement peu utilisés, en comparaison avec d'autres méthodes d'apprentissage statistique, comme nous allons le voir dans la sous-partie suivante.

2.3 Méthodes utilisées pour la prédiction de la qualité dans l'industrie

Nous allons nous intéresser aux différentes méthodes utilisées pour prédire l'état de sortie des processus dans le secteur manufacturier, dans le secteur des services, ainsi que dans les systèmes éducatifs.

2.3.1 Domaine manufacturier

La fouille de données a été appliquée dans le domaine manufacturier depuis les vingt dernières années (Harding, Shahbaz, Srivinas, & Kusiak, 2006). La majorité des applications en qualité sont relatives à des analyses prédictives. De nombreuses méthodes ont été développées, nous allons citer les principales.

Les industries ayant le plus investit dans la prédiction de la qualité sont les industries des produits électroniques, et les industries de métallurgie (Koskal, Batmaz, & Tetik, 2011). La raison est qu'il s'agit d'industries lourdes, dont les processus peuvent être très longs et couteux. Ces derniers sont souvent informatisés, et remontent donc des données en permanence sur l'état des variables de processus, puis sur la qualité finale. Il s'agit donc d'un domaine propice à l'application d'outils d'apprentissage statistique.

Un cas d'application important est la métrologie virtuelle. Celle-ci consiste à prédire des propriétés de qualité sur les semi-conducteurs en cours de production, à partir de données relatives à l'équipement de production, comme la température des fours, ou la pression, de manière à éviter le contrôle qualité physique, qui se révèle très couteux (Khan & Moyne, 2007). Différentes méthodes d'apprentissage statistique ont été appliquées (Tilouche , Bassetto, & Partovi-Nia, 2014). Des prédictions précises ont pu être obtenues, car la corrélation entre les données d'équipement et la qualité finale des produits est grande.

Toujours dans le domaine des semi-conducteurs, Weiss (Weiss, Dhurandhar, & Baseman, 2013) a développé une méthode pour prédire la consommation d'énergie (associée à la qualité finale) des puces produites par une usine, à partir de données recueillies pendant le processus, relatives à des mesures physiques, lithographiques et électriques. L'objectif étant d'obtenir une prédiction avant d'atteindre 50% de la réalisation du processus, de manière à permettre une action avant qu'il ne soit trop tard. Malgré la difficulté de la tâche, liée principalement à un nombre peu élevé de données disponibles, et contenant une quantité importante de valeurs manquantes, une justesse de 80% a pu

être obtenue, ce qui est encourageant pour l'utilisation de techniques similaires en prédiction de qualité.

Dans les domaines de la métallurgie, les méthodes de prédiction peuvent s'intéresser à des caractéristiques de lissage de surface, lors des opérations de tournage (Tsai, Chen, & Lou, 1999), ou bien de qualité de soudage, ou de moulage (Ali & Chen, 1999).

Différentes méthodes d'apprentissage statistiques sont employées.

En ce qui concerne la prédiction par régression, les deux méthodes les plus utilisées sont la régression linéaire multivariée, ainsi que les réseaux de neurones (Koskal, Batmaz, & Tetik, 2011). Cette dernière est reconnue pour fournir des prédictions justes, mais en créant des modèles difficilement, voire impossibles à comprendre par les experts du processus en question.

Par rapport à la prédiction par classification, les deux méthodes les plus utilisées sont les arbres de décision, ainsi que les réseaux de neurones (Koskal, Batmaz, & Tetik, 2011). Si les premiers fournissent souvent des résultats souvent moins performants que les seconds, ils sont cependant reconnus pour retourner des modèles qui peuvent être compris par leurs utilisateurs.

Un point important est que très peu de réseaux bayésiens ont été appliqués pour prédire la qualité de processus de production manufacturiers, sauf un cas particulier et simple, appelé le Classificateur Bayésien Naïf (Perzyk, Biernacki, & Kochanski, 2005). Il s'agit d'un réseau bayésien qui néglige les relations entre les variables d'entrée du modèle (variables observables). Bien que simple, ce modèle est reconnu pour fournir des classifications qui se révèlent souvent relativement intéressantes (Domingos & Pazzani, 1997).

2.3.2 Dans Les Industries De Services

Nous avons choisi de séparer le secteur éducatif, qui sera traité dans la partie suivante, du reste du secteur tertiaire.

Le secteur des services est extrêmement large, et nous nous limiterons à une analyse globale des applications de prédiction dans le domaine.

Il existe de très nombreuses applications de la fouille de données dans le secteur tertiaire. Cependant, très peu de ces applications s'intéressent au schéma suivi dans ce mémoire, dans le

sens où nous cherchons à prédire la qualité de sortie de processus à partir d'observations recueillies durant son déroulement.

La plupart des applications dans le tertiaire sont des analyses descriptives, cherchant par exemple à regrouper des clients d'une entreprise, de manière à optimiser les dépenses en marketing. Ces analyses descriptives sortent du cadre du mémoire.

Une méthodologie particulière appelée Process Mining a été développée (Tiwari, Turner, & Majeed, 2008). Il s'agit en fait d'une application du Data Mining aux processus d'affaires. Les objectifs restent les mêmes : découvrir des formes dans les bases de données des processus, de manière à transmettre de l'information pour améliorer les processus. Cependant là encore, les applications se focalisent plus sur la découverte de formes au sein des processus, que sur la prédiction de la qualité de sortie.

Ainsi, contrairement au secteur manufacturier, les services semblent accorder moins d'importance à la prédiction de qualité finale des processus à partir de variables observées lors du processus qu'à des analyses descriptives

Cependant, des analyses prédictives se font, en particulier dans les domaines des assurances, à partir de données relatives au passé des clients (historique de remboursement par client, conditions sociales…). Ceci est également le cas en finance, où l'on utilise les séries temporelles pour prédire l'évolution du cours des actions. Mais nous allons considérer que ces analyses prédictives diffèrent de notre cadre, qui se focalise sur la prédiction de la qualité en fin de processus.

En ce qui concerne les réseaux bayésiens, ces derniers sont pratiquement inutilisés dans les processus d'affaires du secteur tertiaire, sauf dans un cas particulier : l'industrie des logiciels (Tosun, Bener, & Akbarinasaji, 2015). Ils ont été utilisés ces vingt dernières années pour modéliser les processus de conception des logiciels, de manière à pouvoir prévoir la qualité finale des processus de conception, en tenant compte de l'incertitude, et des connaissances des experts des processus.

2.3.3 Domaine Éducatif

Bien que le domaine éducatif s'éloigne fortement du domaine industriel, nous allons, dans le cadre de ce mémoire, supposer qu'il peut être considéré comme faisant partie du domaine du contrôle des processus, dans le sens où il y a transformation d'étudiants entrants en étudiant sortants, avec

des variables observables relevées durant le déroulement de la formation (par exemple les notes). Il s'agit ainsi d'un service offert aux étudiants, où la qualité sera donc définie par la satisfaction des étudiants pour le service reçu. A noter que, dans le cadre du domaine éducatif, l'indicateur mesuré est habituellement considéré comme une mesure du niveau de l'étudiant, ou des compétences acquises. Mais dans le cadre de ce mémoire, nous faisons l'hypothèse (forte) que les indicateurs mesurés tels que les notes des étudiants rendent compte de la satisfaction de l'étudiant pour le service reçu, bien que cela soit discutable.

Une méthode de fouille de donnée dédiée au domaine éducatif a été développée, il s'agit de l'Educational Data Mining (EDM). Il s'agit de découvrir des formes pouvant transmettre de l'information sur la façon dont les étudiants apprennent, à partir de bases des bases de données des universités ou des systèmes scolaires (Romero & Ventura, 2010).

L'EDM porte un intérêt particulier à l'analyse prédictive, notamment la prédiction du succès des étudiants en cours de formation, ainsi que la prédiction des échecs et des abandons, à partir des données disponibles sur le processus de formation en cours. Nous retrouvons ainsi divers algorithmes d'apprentissage statistique utilisés dans le domaine manufacturier, cherchant à prédire la note finale d'un étudiant donné, à partir de données telles que les notes obtenues par le passé, les conditions sociales des étudiants... Des méthodes basées sur une grande diversité d'algorithmes ont été appliquées, en particulier les réseaux de neurones, les régressions linéaires, les arbres de décision (Delen, 2010).

L'EDM s'intéresse à la modélisation des comportements des étudiants lors de leur apprentissage, en appliquant des algorithmes d'apprentissage machine sur des données réelles ou générées (Besheti & Desmarais, 2015), de manière à prédire les performances des étudiants. Ces dernières concernent souvent les performances lors des questions d'examens. Ainsi, les travaux s'intéressent par exemple à prédire l'habilité des étudiants à répondre à des questions de fins d'examens, sachant les réponses données aux premières questions. La modélisation peut se focaliser sur le développement de méthodes, comme l'utilisation des matrices dites « Q », permettant de mettre en évidence les liens entre les questions d'examens (Items) et les connaissances requises pour y répondre (Skills) (Desmarais, Naceur, & Behsheti, 2012).

Contrairement aux deux domaines précédents, les réseaux bayésiens ont été utilisés à plusieurs reprises pour prédire la réussite des étudiants. Bekele et Menzel(Bekele & Menzel, 2005) ont utilisé

les réseaux bayésiens pour prédire le succès d'étudiants, en les classifiant en trois classes : « Satisfaisant », « Non Satisfaisant », « Plus que satisfaisant ». Les variables d'entrée du modèle étaient des informations relatives aux vies personnelles et aux conditions sociales des étudiants, mais aussi des indicateurs tels que le niveau d'anglais et de mathématiques. Deux ans plus tard, (Haddawy, Thi, & Hien, 2007) ont utilisé les réseaux bayésiens pour prédire la valeur de la moyenne cumulée d'étudiants internationaux postulant pour rentrer à l'Asian Institute of Technology. Les variables d'entrée étaient principalement des informations personnelles sur les étudiants, telles que la nationalité ou le sexe.

Plus récemment, Sharabiani les a appliqués pour prédire les notes des étudiants à des cours clefs du programme d'ingénierie de l'université de Chicago, en se basant encore sur des facteurs sociaux et personnels (Sharabiani, 2014).

Toutes les applications des réseaux bayésiens en EDM ont donné des résultats satisfaisants par rapport à des benchmarks d'algorithmes de prédiction. Tous ont choisi les réseaux bayésiens pour leur habilité à travailler avec des situations incertaines, qui sont caractéristiques des processus de formation d'étudiants.

Cependant, un point important reste à préciser par rapport à la prédiction des succès des étudiants. Les travaux effectués ont cherché à prédire la moyenne cumulée des étudiants à la fin du cursus, pour une session donnée, ou pour un cours donné, en se basant principalement sur des informations relatives aux vies personnelles des étudiants en questions, par exemple le sexe, la nationalité, ou les notes antérieures au processus.

Dans notre cas d'étude, nous n'avons pas accès à des informations relatives aux vies personnelles et aux facteurs sociaux concernant les étudiants en question, ni à leurs notes antérieures à leur baccalauréat à Polytechnique (par exemple au CEGEP). Nous avons seulement accès aux moyennes par session, au département, ainsi qu'au nombre de crédits pris pour chaque étudiant. Le cadre de l'étude est donc différent.

La possibilité de prédire la note moyenne obtenue pour un étudiant donné lors de la seconde partie de son baccalauréat dépendra de la présence de « trajectoires de moyennes par session » (formes, ou patterns) dans la première partie du baccalauréat. Bien que les données de Polytechnique n'aient pas encore étés étudiées lors de projets scientifiques, nous avons effectué quelques recherches dans la littérature par rapport à l'existence de telles trajectoires. (Grove & Wasserman, 2004) ont étudié

les trajectoires d'étudiants au baccalauréat d'universités américaines, à partir de données de cinq cohortes. Ils ont remarqué que la trajectoire moyenne des étudiants prenait une forme de V : une chute de moyenne cumulative de la première à la deuxième session, puis une augmentation constante de la deuxième à la dernière session de baccalauréat. Ils attribuent ce cycle à plusieurs causes : les abandons, mais aussi la participation à des comités et fraternités, ainsi que de multiples autres facteurs, tels que la prise de maturité des étudiants, ou le gain d'intérêt dans les études à mesure que les sujets étudiés se précisent. (Orr, 2011) a effectué une étude similaire, mais en se focalisant sur les étudiants en génie. Ils décrivent une trajectoire similaire à celle montrée par Grove : une forme en V.

Ainsi, les articles de la littérature indiquent la présence de formes dans l'évolution des moyennes dans le temps. Bien que les données de Polytechnique soient différentes, cela signifie tout de même que, de manière générale, les données ne suivront probablement pas un comportement chaotique.

2.4 Synthèse de la revue de littérature

La revue de littérature a permis de présenter les réseaux bayésiens, ainsi que de montrer le contexte de la prédiction de la qualité de sortie des processus.

Les réseaux bayésiens sont populaires pour proposer des modèles compréhensibles par les experts des processus en question, de par leur aspect graphique, qui permet de diminuer l'effet « boite noire », qui est présent avec des méthodes telles que les réseaux de neurones, ou les machines à vecteurs de support.

De plus, leur aspect probabiliste leur donne l'avantage d'être adaptés pour les situations comportant de l'incertitude, c'est-à-dire, lorsque de l'information manque pour prédire la valeur d'une variable d'intérêt. L'incertitude est particulièrement présente pour des processus impliquant des êtres humains, comme les processus de formation d'étudiants, qui ont pour caractéristique d'être difficilement prévisibles.

La prédiction de qualité a principalement été développée et appliquée dans les secteurs manufacturiers fortement automatisés, tels que la production de semi-conducteurs. De nombreux travaux se sont focalisés sur l'application d'algorithmes d'apprentissage machine, pour prédire en avance la qualité finale des produits en cours de production à partir des indicateurs disponibles.

Cette prédiction de qualité est également utilisée dans d'autres domaines, tels que l'industrie des logiciels.

Les systèmes éducatifs s'intéressent aussi à la prédiction du succès de leurs étudiants, que nous assimilons ici à la satisfaction de leurs clients, soit la qualité du service offert. Un point important doit pourtant être soulevé. Les méthodes utilisées pour prédire le succès d'étudiants sont principalement basées sur des informations d'ordre social ou personnel, propres à la vie de l'étudiant avant le processus de formation. Or, dans notre cas d'étude, nous avons seulement accès à des informations qui décrivent l'état du processus en cours (notes par session, nombre de crédits et Département), et non à des informations personnelles, ou relatant des caractéristiques avant le commencement du processus.

D'autre part, parmi les méthodes utilisées en qualité, très peu sont relatives à l'utilisation de réseaux bayésiens.

L'une des raisons est que pendant longtemps, très peu d'outils étaient disponibles pour permettre une intégration de réseaux bayésiens avec des logiciels (Lee & Abbott, 2003). Cependant, avec la montée de la capacité de calcul des ordinateurs ces dix dernières années, des moteurs d'inférence bayésienne ont pu être développés, permettant ainsi leur utilisation. L'un de ces moteurs est le logiciel ProBT, que nous avons utilisé dans notre système de prédiction.

Une autre conclusion importante qui peut être tirée de la revue de littérature est que la plupart des systèmes de prédiction proposés dans le domaine de la qualité sont des systèmes ressortant une prédiction unique, c'est-à-dire non probabiliste. Si des méthodes probabilistes ont été utilisées (comme le classificateur bayésien naïf dans le domaine manufacturier, ou des réseaux bayésiens dans les systèmes éducatifs), la plupart évaluent la justesse des valeurs prédites, et non pas des distributions de probabilités prédites.

Dans des cas de processus dont l'issue est incertaine comme celui étudié dans ce projet, il peut être intéressant de fournir en tant que prédiction non pas une valeur ou classe unique, mais une distribution de probabilités sur l'ensemble des valeurs possibles, de manière à pouvoir visualiser les différents risques encourus par le processus en cours d'une part, et à avoir une idée de la « confiance » à accorder à la prédiction fournie par le système d'autre part.

Pour cela, il est intéressant d'évaluer l'estimation probabiliste fournie par le système de prédiction, plutôt que simplement la justesse de la classe ou la valeur fournie. C'est ce que nous proposons d'évaluer dans la partie test de ce mémoire.

CHAPITRE 3 PROPOSITION

Ce chapitre présente la solution proposée pour répondre à la problématique, d'un point de vue général : fournir une prédiction probabiliste de l'état de sortie à partir des variables observées, en utilisant les réseaux bayésiens. Le chapitre 4 s'intéressera à l'application de cette proposition sur le cas d'étude.

Nous allons présenter ici une description générale du fonctionnement du système de prédiction avant de détailler chaque étape séparément.

Le système s'insère de la manière suivante au sein du processus de production :

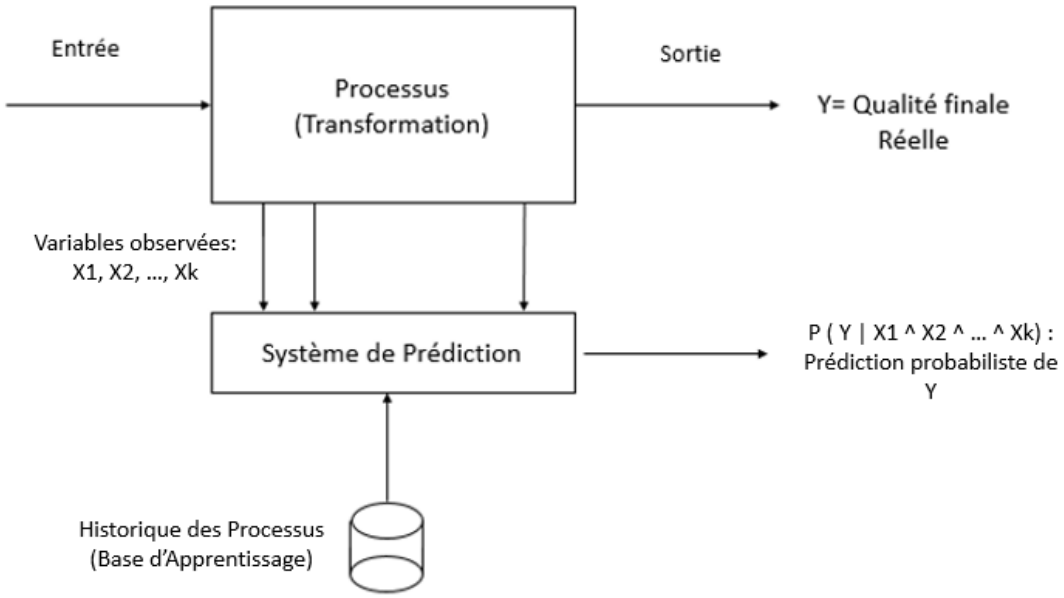


Figure 3-1: Positionnement du système dans son environnement. Inspiré de (Khan, Moyne, & Tilbury, An Approach for Factory-Wide Control Utilizing Virtual Metrology, 2007)

.Notre proposition est ainsi basée sur la méthode appelée métrologie virtuelle, appliquée en particulier dans les industries automatisées des semi-conducteurs. Elle se positionne de la même manière au sein du processus de production. Elle peut ainsi être vue comme généralisation de métrologie virtuelle à la prédiction de la qualité.

Il s'agit donc, comme nous l'avons vu en introduction, de prédire en temps réel la qualité finale du processus, de manière probabiliste, à partir des variables observées lors du processus.

La particularité de la proposition est l'insistance sur l'aspect probabiliste de la prédiction, lui permettant de s'intéresser aux processus où l'incertitude est importante, et ainsi de permettre une aide à la décision aux experts.

Nous allons dans ce chapitre, désigner par X1, …, Xk les variables observées lors du déroulement du processus, et par Y la qualité finale en sortie.

3.1 Fonctionnement global

La figure 3.2 présente le fonctionnement global du système de prédiction proposé.

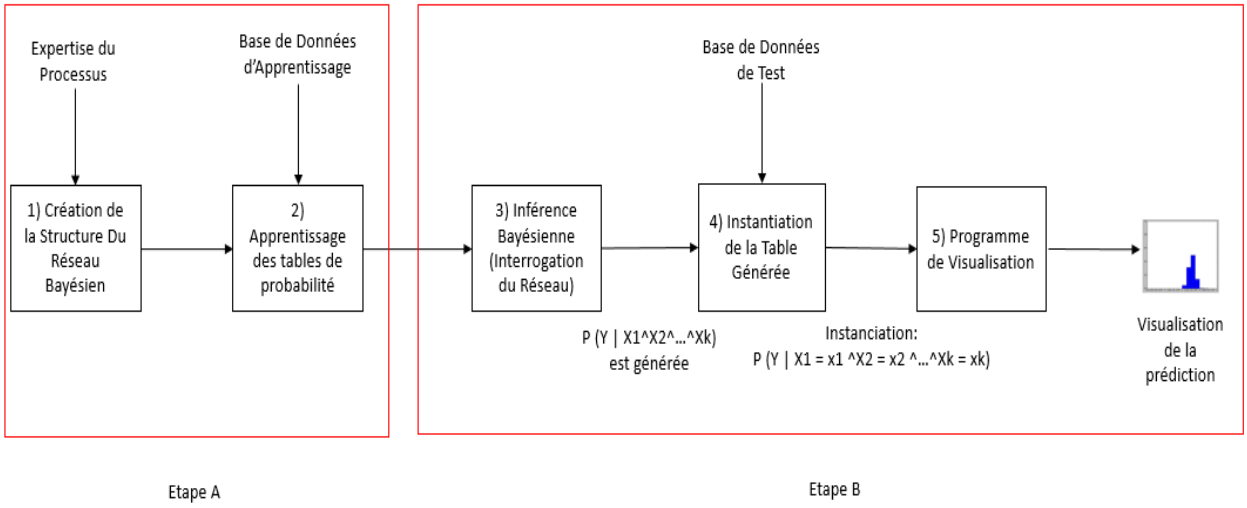


Figure 3-2: Fonctionnement Global du Système

Le système peut au premier abord se diviser en deux grandes étapes : l'étape A de l'apprentissage, et l'étape B de l'usage.

Chacune de ces étapes se décompose elle-même en sous-étapes.

L'apprentissage consiste à créer le réseau bayésien, et à lui faire apprendre les paramètres nécessaires pour l'inférence. Il s'agit d'une étape qui doit être considérée pour chaque cas d'application, en fonction des connaissances d'experts et des données disponibles. La première sous-étape consiste à apprendre la structure, c'est-à-dire déterminer les corrélations entre les variables du phénomène à modéliser. S'il existe des méthodes pour apprendre automatiquement, à partir d'une base de données, la structure d'un réseau bayésien, nous allons, dans le cadre de ce projet, nous limiter à une création manuelle du réseau. Il s'agit de l'aspect qualitatif des réseaux

bayésiens, qui sont le plus souvent, comme nous l'avons vu, déterminés par les experts des phénomènes en question.

La seconde sous-étape consiste à apprendre les paramètres qualitatifs, c'est-à-dire les tables de probabilités des différents nœuds du réseau bayésien. Cette étape se fera, dans le cadre de notre projet, par apprentissage à partir d'une base de données.

La seconde étape du fonctionnement du système concerne l'usage. Il s'agit du système lorsqu'il est effectivement déployé sur le processus, suite à l'apprentissage. Cela implique l'interrogation du réseau bayésien qui a été appris à l'étape précédente, de manière à obtenir une prédiction probabiliste de la variable d'intérêt. C'est l'étape où l'inférence bayésienne intervient, en calculant la distribution de probabilité a posteriori P (Y | X1 ^ … ^ Xk) à partir de la distribution conjointe P (Y ^ X1 ^ X2 ^ … ^ Xk).

Ceci est le point principal qui différencie les réseaux bayésiens des méthodes d'apprentissage statistiques non probabilistes.

Un système d'apprentissage statistique non probabiliste cherchera à déterminer la fonction reliant la matrice des variables d'entrée du processus à la variable de sortie S2.

Le réseau bayésien, qui est un modèle probabiliste, a un fonctionnement différent. Il s'intéresse à la distribution conjointe reliant toutes les variables du processus :

$$P (Y \wedge X1 \wedge X2 \wedge \ldots \wedge Xk)$$

Une fois la distribution conjointe apprise, il l'utilise avec l'inférence bayésienne pour déterminer la distribution P (Y | X1 ^ … ^ Xk) a posteriori de Y.

Ainsi, il n'y a pas, à proprement parler, de véritables « sorties » ou « entrées » dans un réseau bayésien, mais seulement des variables corrélées (ou non), qui peuvent être inférées, sachant les valeurs d'autres variables. Nous continuerons cependant le terme « variables d'entrées » et « variables de sortie » pour rester en phase avec la littérature sur la prédiction de la qualité.

Toutes les étapes de la création du système proposé sont faites à partir de la librairie Python ProBT, tirée de : http://emotion.inrialpes.fr/BP/spip.php?rubrique6 .

Les parties suivantes présentent en détail chaque sous-partie du système de prédiction.

3.2 Conditions de fonctionnement

Pour que l'outil puisse fonctionner, c'est-à-dire fournir des prédictions informatives, nous avons besoin :

- Qu'il y ait des formes dans les données d'entrées, et une relation entre l'entrée et la sortie. Les données doivent être identiquement et indépendamment distribuées entre les exemples. Cela signifie que pour que le système puisse reconnaitre des tendances, il faut que des tendances similaires aient été observées durant l'apprentissage.
- Qu'il y ait un certain nombre de données (d'exemples) pour permettre un apprentissage efficace. L'effet de ce paramètre sera mis en évidence par la suite dans le mémoire.
- Il s'agit d'apprentissage supervisé. Les données de sortie doivent donc être explicites. Un prétraitement doit donc être effectué si la base de données « brute » ne présente pas les données de cette manière. Le tableau ci-dessous illustre le format de la base de données, que ce soit pour le test ou l'apprentissage.

Table 3-1: Format de la base de données

Etudiant	X1	X2	...	Xk	Y
1					
2					
...					
n					

Variables Observées | Variable d'Intérêt

3.3 Création de la Structure du Réseau Bayésien

La création de la structure est une étape primordiale de l'apprentissage. C'est elle qui permet d'indiquer les corrélations entre les variables que le modèle doit prendre en compte pour pouvoir effectuer des prédictions.

La littérature parle également de « décomposition de la distribution conjointe » (Bessière, Mazer, Ahuactzin, & Mekhnacha, 2013).

Cela signifie qu'un réseau bayésien est habituellement utilisé pour décomposer, c'est-à-dire simplifier la distribution conjointe des variables du modèle, de manière à faciliter les calculs d'inférence qui peuvent vite devenir extrêmement complexes (NP difficiles).

L'avis d'expert permet de spécifier quelles variables interagissent avec quelles autres, de manière à simplifier efficacement le réseau.

Dans le cadre du projet, nous nous intéressons (uniquement) à la fusion d'information, c'est-à-dire que nous cherchons à obtenir une estimation probabiliste de la variable d'intérêt Y sachant les valeurs des variables de processus en entrée. Cette fusion d'information permet ainsi de faire de la reconnaissance de formes, de manière probabiliste (Bessière, Mazer, Ahuactzin, & Mekhnacha, 2013).

Dans ce chapitre, à titre d'exemple illustratif, nous allons nous limiter au réseau le plus basique permettant de fusionner l'information : le réseau bayésien naïf. Ce modèle est toujours le même, et ainsi ne requiert pas d'expertise pour être créé.

Il s'agit d'un réseau bayésien reliant directement toutes les variables d'entrée du processus à la variable de sortie, appelé naïf du fait qu'il ne prend pas en compte les corrélations entre les variables d'entrée : il suppose l'indépendance conditionnelle de ces variables d'entrée sachant la variable de sortie Y.

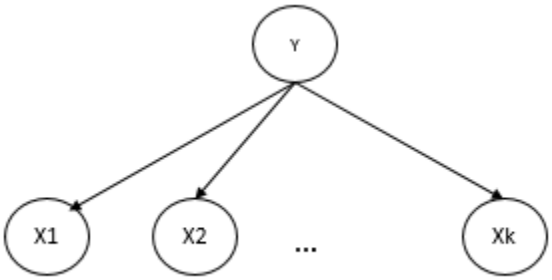


Figure 3-3: Réseau Bayésien naïf

La distribution conjointe décomposée associée est la suivante :

$$P\ (Y \wedge X1 \wedge X2 \wedge \ldots \wedge Xk\ ) = P\ (Y) \times \prod_{i=1}^{i=k} P\ (Xi\ |Y)$$

Chaque terme de la décomposition correspond à une table de probabilités qui sera apprise de la base de données d'apprentissage.

Deux types de distribution sont présents : les distributions de probabilités a priori, comme P(Y), et les distributions conditionnelles, comme P (Xi |Y).

De manière générale, le nombre de paramètres d'une table P (A | B1 ^ … ^ BN) se définit par :

$$\text{Card(A) x } \prod_{i=1}^{N} Card(Bi)$$

Où Card(V) représente le nombre de valeurs que peut prendre la variable V. Ainsi, plus le nombre de nœuds parents et élevé, plus le nombre de paramètres des tables de probabilités de la variable considérée augmente. Un réseau ayant trop de connexions sera confronté à la malédiction de la dimensionnalité : besoin d'un très grand nombre de données, besoin d'une grande capacité de mémoire, et calculs couteux. Il s'agira ainsi de bien choisir la décomposition, permettant de limiter le biais d'une part, mais de limiter la complexité d'autre part. Un autre paramètre important est le cardinal des variables, qui sera abordé dans la sous-partie suivante.

Un point peut être soulevé, par rapport au sens des flèches. Le réseau bayésien proposé ici n'est pas un réseau causal, dans le sens où l'on ne s'intéresse pas à proposer un modèle dont les variables parents ont un lien de causalité avec les variables pointées.

L'objectif ici est de fusionner l'information. Pour se faire, nous avons besoin de décomposer les distributions conjointes de la manière montrée ci-dessus (Bessière, Mazer, Ahuactzin, & Mekhnacha, 2013).

Utiliser un réseau ayant un sens des arcs « causal », par exemple en considérant que les moyennes par session sont les parents (causes) de la moyenne finale, ne permettrait pas de décomposer efficacement la distribution conjointe, dans le sens où la variable S2 à prédire aurait un nombre de paramètres à estimer trop important.

3.4 Apprentissage des tables de probabilités

Une fois la structure du réseau déterminée, la partie quantitative de l'apprentissage doit être faite. Il s'agit d'apprendre les tables de probabilités du réseau, c'est-à-dire les termes de la distribution de probabilité conjointe décomposée, à partir de la base de données.

Pour illustrer nos propos, reprenons l'exemple du réseau bayésien naïf. Comme nous l'avons vu, sa distribution se décompose de la manière suivante :

$$P\ (Y\ \hat{}\ X1\ \hat{}\ X2\ \hat{}\ \dots\ \hat{}\ Xk\ ) = P\ (Y)\ x\ \prod_{i=1}^{i=k} P\ (Xi\ |Y)$$

Dépendamment du processus d'application, les variables peuvent être continues ou catégoriques. Nous allons en premier lieu avoir besoin de discrétiser les variables continues. Ensuite seulement nous pourrons procéder à l'apprentissage depuis la base de données.

3.4.1 Caractérisation et Discrétisation des Variables

Quand on parle de discrétisation des variables, il s'agit de découper les variables continues en sous-intervalles, de manière à ce que l'apprentissage des différentes probabilités puisse se faire à partir de la base de données, en déterminant les différentes fréquences. Cette étape est délicate, considérée comme l'un des principaux points faibles des réseaux bayésiens (Friedman & Goldszmidt, 1996). Chaque variable doit être discrétisée selon un nombre spécifique d'intervalles, à déterminer.

Cependant, il y a nul besoin de discrétiser les variables catégoriques.

La discrétisation définit le nombre d'intervalles dans lequel les variables continues sont divisées. Ces intervalles déterminent le cardinal des variables continues, ils sont donc intimement liés au nombre de paramètres à apprendre des données.

La détermination du nombre d'intervalles se fait en testant différentes valeurs, et en observant l'effet sur la qualité de la prédiction. Cette discrétisation est donc nécessaire pour chaque application sur un nouveau processus.

3.4.2 Apprentissage des tables depuis la base de données

Une fois les variables discrétisées (ou non dans le cas des variables catégoriques), les paramètres, c'est-à-dire les valeurs des probabilités (fréquences) de chaque intervalle doivent être appris.

Là encore, les connaissances d'experts peuvent intervenir. Par exemple, s'il est connu que des variables suivent Loi Normale, il ne reste qu'à déterminer leurs paramètres, à savoir leurs moyennes et leurs écart-types, par la méthode du maximum de vraisemblance.

Dans notre cas, nous considérons que si nous n'avons aucune connaissance a priori sur la forme des distributions de probabilités des différentes variables, nous aurons recours à un apprentissage

non paramétrique, en utilisant une estimation bayésienne des paramètres de la table de probabilités basée sur la formule de Laplace (cas particulier de la distribution de Dirichlet).

L'idée est la suivante. Le système suppose a priori que toutes les valeurs, c'est-à-dire toutes les classes « discrétisées » ont été observées une fois. Ainsi, avant que l'apprentissage de la base de données ne commence, toutes les tables de probabilités sont remplies a priori par une distribution uniforme.

La formule de Laplace permet d'exprimer la valeur de la probabilité Pi de l'élément discrétisé i, c'est-à-dire après observation dans la base de données :

$$\mathrm{Pi} = \frac{1+ni}{\sum_i (1+ni)}$$

Où ni correspond au nombre d'observations de l'élément i.

Cela a deux avantages :

- Tout d'abord, aucune valeur i n'a une probabilité associée Pi nulle. Cela permet d'éviter de se retrouver avec une probabilité pi nulle si la valeur i n'a jamais été observée (ni =0).En effet, le fait d'avoir pi=0 se révèle problématique si, lors du test, la valeur i apparait : la probabilité nulle se répand dans le réseau lors de l'inférence, et aboutit à une estimation probabilité de la sortie correspondant à une distribution nulle.
- Ensuite, cette méthode permet de relativiser l'importance de certaines observations. Citons un exemple, avec une variable aléatoire Vi a dix intervalles. Si la méthode Bayésienne n'est pas utilisée, le fait d'observer une seule fois la valeur i=5 et aucune fois les autres conduira à une probabilité P5 = 1.0, et pi = 0 pour les neuf autres valeurs (figure 3.4). Si la méthode bayésienne est utilisée, avec la formule de Laplace, nous aurons a priori, Pi = 0.1 pour tout i, et a posteriori, P5 = 2/11 = 0.18, et Pi = 1/11 = 0.09 pour les neuf autres valeurs (figure 3.5). Ainsi, la méthode bayésienne permet de limiter l'effet du bruit apporté par certaines observations peu importantes, en particulier lorsque peu de données d'entrainement sont disponibles, ou lorsqu'un nombre important de paramètres doit être appris (par exemple, plusieurs parents pour un même nœud, avec un nombre important d'intervalles).

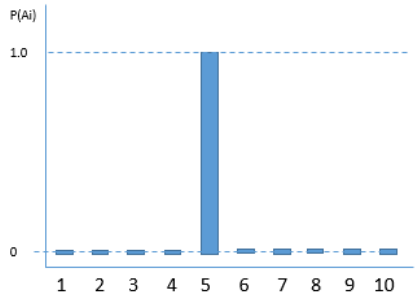


Figure 3-4 : Effet d'une seule observation sans Apprentissage Bayésien sur p(Vi)

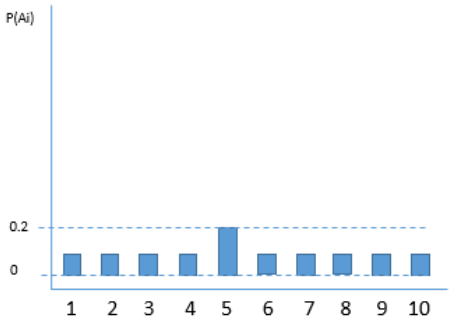


Figure 3-5: Effet d'une seule observation avec la formule de Laplace sur p(Vi)

Ainsi, toutes les tables correspondantes aux termes de la distribution conjointe décomposée seront apprises depuis la base de données par estimation bayésienne avec la formule de Laplace.

3.5 Inférence bayésienne

Une fois la structure du réseau déterminée et les tables de probabilité apprises, le système peut alors procéder à l'inférence, pour déterminer l'estimation de la variable d'intérêt Y. l'inférence bayésienne peut être considérée comme l'interrogation d'une variable du réseau sachant d'autres variables, soit la détermination de la distribution de probabilités a posteriori voulue. Dans notre cas, nous voulons déterminer la distribution a posteriori P(Y | X1^…^Xk).

Sachant la distribution conjointe P(Y ^X1 ^ …^ Xk), la distribution a posteriori P(Y | X1^…^Xk) se calcule, selon l'inférence bayésienne, de la manière suivante :

$$P(Y \mid X1 \wedge \ldots \wedge Xk) = \frac{P(Y \wedge X1 \wedge \ldots \wedge Xk)}{\sum_Y P(Y \wedge X1 \wedge \ldots \wedge Xk)}$$

Le calcul d'inférence est simplifié par la décomposition de la distribution conjointe, selon le réseau naïf dans notre exemple. Les calculs sont effectués de manière automatique par le moteur d'inférence de ProBT.

Le terme au dénominateur est appelé « Constante de marginalisation ». Il s'agit de la somme (intégrale) de la distribution conjointe sur toutes les valeurs possibles prises par la variable inconnue recherchée.

Les probabilités des intervalles de Y sont calibrées par ProBT, de manière à ce que la somme des probabilités soit égale à 1, indépendamment de la taille des intervalles. Ainsi, la distribution a posteriori correspond à une fonction de masse, et non pas à une densité de probabilités. Y est donc toujours perçue comme une variable discrète (discrétisée si elle est continue à l'origine).

Ainsi, le moteur d'inférence génère une distribution a posteriori P(Y | X1 ^ X2 ^ … ^ Xk), qui n'a alors plus qu'à être instanciée pour ressortir une prédiction probabiliste sur Y, sachant les valeurs des variables observées pour le processus en cours.

3.6 Instanciation de la table et prédiction

Pour interroger la table, le système a tout d'abord besoin d'avoir accès aux valeurs des variables observées. Ainsi, il va automatiquement chercher les valeurs de ces dernières dans le fichier csv dit de « test », contenant les valeurs des variables du processus en cours.

Le système instancie les valeurs des variables connues (X1=x1, …, Xk=xk), et ressort de la table de probabilités la distribution a posteriori P (Y | X1=x1 ^ … ^ Xk=xk) correspondante. Il s'agir de l'étape 4. Cette distribution a posteriori (sous forme d'histogramme) correspond à la prédiction.

3.7 Visualisation de la prédiction

Un programme a été développé pour visualiser l'histogramme dans une fenêtre à part. Ce programme est basé sur la librairie de visualisation de données Python Matplotlib (étape 5).

La figure 3.6 présente un exemple de prédiction ressortie par le système, dans le cas où Y est une variable continue entre 0 et 4, qui a été discrétisée en 16 intervalles (cas d'étude, chapitre suivant).

La durée des calculs du nombre de données et des paramètres tels que la discrétisation où la complexité du réseau. Plus d'information à ce sujet sera fourni lors de l'étude du cas d'application, au chapitre 4.

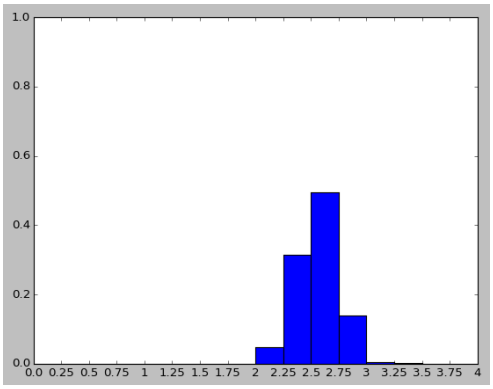


Figure 3-6: Exemple de Prédiction de Y avec un nombre d'intervalles de 16

Nous pouvons voir que la prédiction probabiliste permet de mettre en évidence les valeurs de la qualité de sortie les plus probables, mais également les autres valeurs (moins) probables. Cette représentation permet ainsi de mettre en évidence l'incertitude liée à la prédiction, et de prendre une décision éclairée quant à la performance du processus.

Cette représentation probabiliste fait du système un outil d'aide à la décision. Il ne s'agit pas ainsi de remplacer les responsables de contrôle qualité des systèmes de production par un système agissant de manière automatique sur le processus en cours selon la qualité prévue, mais de les aider à visualiser les différents risques en ce qui concerne l'évolution de la qualité du processus. Cette visualisation trouve un intérêt particulier dans le cas de processus impliquant des humains, qui sont ainsi difficilement prévisibles, et où des décisions ne pourraient pas être prises de manière automatique.

Après avoir présenté le système proposé, nous allons maintenant nous intéresser au cas d'application, qui va permettre de le tester, sur un cas concret.

CHAPITRE 4 CAS D'APPLICATION

4.1 Contexte

Nous proposons d'utiliser notre outil pour fournir une prédiction probabiliste de la moyenne cumulative qu'un étudiant donné est susceptible d'obtenir lors de la seconde partie de son baccalauréat sachant le département, les notes obtenues et le nombre de crédits pris lors de la première partie du baccalauréat.

Comme nous l'avons précisé aux chapitres précédents, nous considérons que la formation des étudiants est un processus.

Nous considérons que les variables d'entrée (nombre de crédits, notes, département) correspondent aux variables observables durant la transformation.

Aussi, nous allons supposer que la qualité de sortie du processus de formation, c'est-à-dire la satisfaction des clients (étudiants) pour le service reçu, est descriptible par la moyenne cumulative obtenue à la fin du baccalauréat. Il s'agit d'une hypothèse forte, étant donné qu'un étudiant peut avoir une moyenne cumulative élevée tout en étant pas satisfait de son expérience, mais ceci est l'unique indicateur disponible pour évaluer la qualité de la formation, avec les données disponibles.

Nous allons vérifier s'il existe des patterns temporels dans les données des étudiants, permettant la prédiction de la qualité de sortie.

L'idée est d'utiliser la prédiction de la moyenne d'un étudiant donné, de manière à pouvoir l'aider ou lui donner des conseils, afin d'anticiper d'éventuels échecs, et améliorer la satisfaction des étudiants.

Pour cela, nous avons besoin d'avoir des prédictions fiables. C'est ainsi la performance du système en matière de prédiction que nous allons évaluer.

4.1.1 Processus de Formation

Le baccalauréat à Polytechnique Montréal dure en moyenne quatre années. Dans ce travail, nous allons considérer que cette formation se fait en 12 sessions, soit 3 sessions par année, en prenant en compte la session d'été.

Au cours de leur formation, les étudiants reçoivent une note moyenne pour chaque session, qui est enregistrée dans la base de données. Sont également enregistrés le nombre de crédits pris par l'étudiant pour chaque session, ainsi que le département de l'étudiant en question.

A la fin de sa formation, l'étudiant a cumulé 12 sessions, et possède une moyenne cumulative finale, qui reflète (théoriquement) son niveau, ou la satisfaction de son expérience à Polytechnique.

Dans ce travail, nous allons donc considérer que l'on connait, pour chaque étudiant, les notes obtenues (notées Ai), le nombre de crédits pris (noté Ci) et le département (D), pour les 6 premières sessions de son baccalauréat.

Les données réelles que Polytechnique nous a transmises proviennent de la cohorte de l'Automne 2008 (donc jusqu'à l'Été 2012).

Le tableau 4.1 résume la signification des variables utilisées par le système.

Table 4-1: Description des Variables

Variable	Signification	Valeurs	Type
D	Département	{1, …, 11}	Catégorique
Ai	Note moyenne obtenue à la session i	[0,4]	Réel
Ci	Nombre de Crédits pris à la session i	{0, …, 18}	Entier
S2	Moyenne Cumulative lors de la Seconde partie du Baccalauréat	[0,4]	Réel

Dans la base de données, la première session (i=1) correspond à la session d'Automne 2008, la deuxième (i=2) correspond à la session d'Hiver 2009, la troisième (i=3) la session d'Été 2009, et ainsi de suite. Ainsi, l'effet des saisons apparait implicitement dans les données.

Nous cherchons à déterminer la moyenne cumulative obtenue lors de la seconde partie du baccalauréat pour la raison suivante.

La moyenne cumulative finale du processus se définit par :

$$Sfinale = (A1 \times C1 + \ldots + A6 \times C6 + A7 \times C7 + \ldots + A12 \times C12) \times 1/ (C1 + \ldots + C12)$$

Or, l'objectif est de connaitre la moyenne cumulative finale connaissant les valeurs d'A1, C1, … A6, C6, obtenues lors de la première partie du baccalauréat.

Ainsi, il est inutile de prédire une valeur contenant des termes relatifs à la première partie du baccalauréat, qui sont déjà connus pour un étudiant en 6ème session.

Nous nous intéressons donc à prédire uniquement la partie encore inconnue de la moyenne cumulative finale, S2= (A7 x C7 + … + A12 x C12) / (C7 + … + C12).

Nous choisissons de prédire à partir de six sessions, car nous avons besoin d'une part d'un certain nombre d'informations (donc de sessions passées) pour pouvoir espérer distinguer des tendances. Plus nous avons d'information, plus la prédiction devrait être précise. D'autre part, le but de la prédiction est de permettre une action de la part de l'école, de manière à aider un étudiant donné à affronter d'éventuelles difficultés prédites. Plus l'on avance dans le processus, plus il devient difficile pour l'étudiant d'agir sur la moyenne cumulative totale obtenue. Ainsi, nous choisissons d'effectuer la prédiction à la moitié du processus, comme l'a choisi Weiss dans le cas de la prédiction de la qualité finale lors de la production de puces électroniques (Weiss, Dhurandhar, & Baseman, 2013).

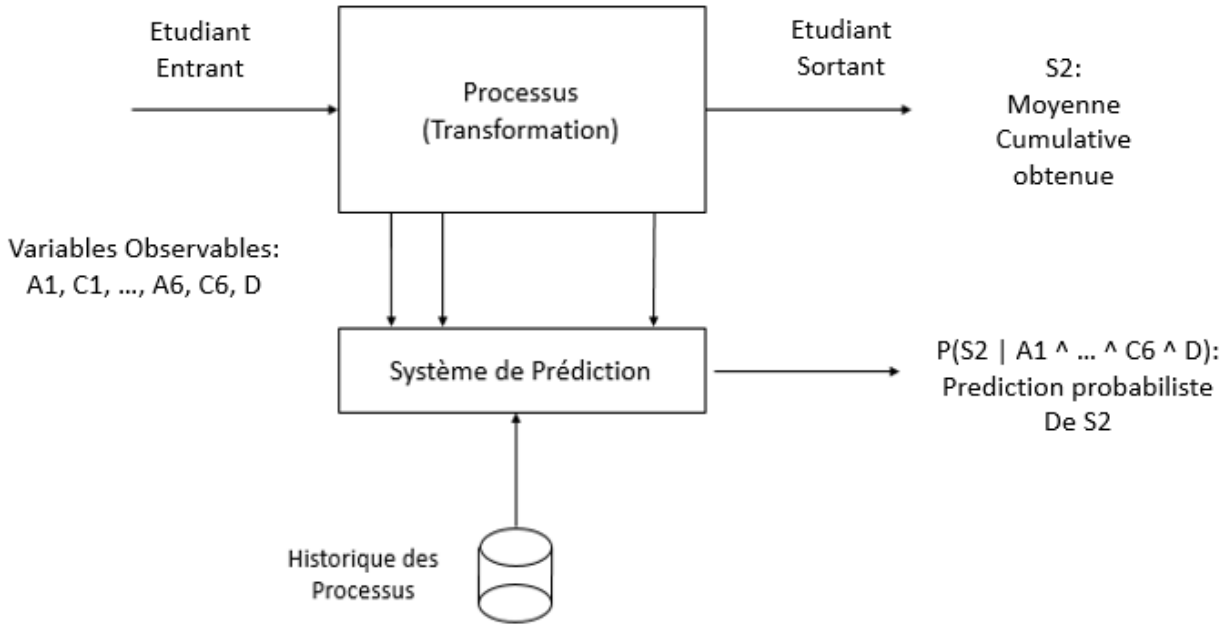


Figure 4-1: Le système et le processus

La figure 4.1 décrit le positionnement du système de prédiction dans son environnement. Les variables observables correspondent aux informations disponibles lorsque la prédiction doit être effectuée, c'est-à-dire à la moitié du processus, à la fin de la sixième session.

Un apprentissage est effectué à partir de l'historique des processus, c'est-à-dire les étudiants des cohortes précédentes.

Une fois l'apprentissage effectué, le système récupère les valeurs des variables d'entrée du processus en cours, interroge le réseau bayésien avec ces données, et ressort une prédiction probabiliste de la sortie.

4.1.2 Données du processus et Prédiction

La base de données de l'historique des processus a un format suivant le tableau 4.2.

Les lignes du tableau représentent les étudiants (les exemples). Les colonnes allant de D à C6 représentent les variables observées. La Colonne S2 représente les moyennes cumulées obtenues, pour chaque étudiant, à la seconde session de son baccalauréat, soit la sortie.

Le système de prédiction utilisé doit discrétiser les variables continues, donc en particulier S2. Ainsi, la variable de sortie ne sera pas perçue, dans le cadre de ce travail, comme une variable continue, mais comme une variable discrétisée. Nous allons donc considérer qu'il s'agit de classification (probabiliste), et non pas de régression.

Une fois l'apprentissage effectué, lors de l'utilisation de l'outil, les valeurs des variables de processus seront connues pour un étudiant donné, et la variable S2 sera la variable à prédire.

Table 4-2: Format de la Base de données d'apprentissage

Etudiant	D	A1	C1	A2	C2	...	A6	C6	S2
1									
2									
...									
n									

Variables de Processus (d'Entrée) Variable d'Intérêt

Dans ce projet, nous avons dans un premier temps généré des données, contenant des formes, de manière à tester l'habilité du système à prédire la sortie, en fonction de paramètres maitrisés.

Nous avons dans un second temps effectué une validation croisée sur les données réelles (480 étudiants), pour tester l'habilité à effectuer des prédictions dans les conditions réelles.

4.2 Implémentation du système

Pour implanter le système sur le processus étudié, nous suivons la démarche décrite dans le chapitre précèdent.

4.2.1 Création de la structure du réseau bayésien

Deux types de réseaux seront comparés lors du test : le réseau naïf et le réseau naïf augmenté.

Nous allons commencer avec la structure de base pour fusionner l'information avec les réseaux bayésiens est décrite comme la fusion dite « naïve » (Bessière, Mazer, Ahuactzin, & Mekhnacha, 2013).

Il suppose l'indépendance conditionnelle de ces variables d'entrée sachant la variable de sortie S2.

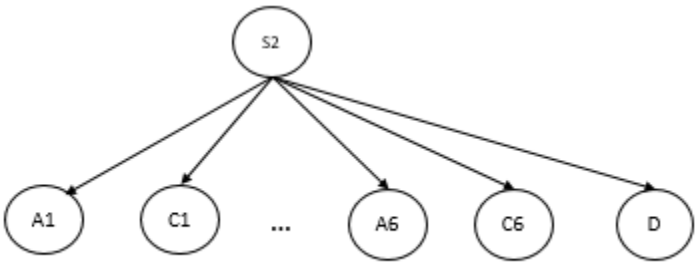


Figure 4-2: Réseau Bayésien Naïf

Il s'agit d'une hypothèse forte, souvent fausse, qui introduit du biais dans les prédictions.

Cependant, malgré son aspect naïf, il s'est avéré que ce réseau fournisse des résultats étonnamment bons dans le cas de la classification, c'est-à-dire quand il est utilisé pour prédire la valeur prise par la classe de sortie (Domingos & Pazzani, 1997).

Par contre, s'il est considéré comme un bon modèle pour la classification, il s'est révélé être un modèle fournissant des estimations probabilistes de la valeur de sortie souvent biaisées, du fait qu'il néglige les corrélations entre les variables d'entrée (Domingos & Pazzani, 1996).

La décomposition du réseau naïf s'écrit de la manière suivante :

$$P (D \wedge A1 \wedge C1 \wedge A2 \wedge C2 \wedge \ldots \wedge A6 \wedge C6 \wedge S2) =$$

$$P(S2) \times P(D \mid S2) \times \prod_{i=1}^{6} [P(Ai|S2) * P(Ci|S2)]$$

L'avantage principal du réseau bayésien naïf réside dans le fait que la décomposition proposée réduit considérablement le nombre de paramètres à apprendre, car les tables de probabilités comportent au maximum une seule variable dite « parent ». Il permet ainsi de limiter très fortement la malédiction de la dimensionnalité, et donc d'une part, d'être relativement efficace dans des situations où le nombre de données est limité, et d'autre part, de simplifier grandement les calculs d'inférence.

Le nombre de paramètres nécessaires pour ce modèle est le suivant :

$$\text{Card(S2) + Card (D) x Card (S2) + } \prod_{i=1}^{6}[Card(Ci) * Card(S2)]$$

De manière à limiter le biais introduit par la simplification du réseau naïf, nous proposons un réseau dit « naïf augmenté » (Cheng & Greiner, 1999). Il s'agit toujours de fusionner l'information d'entrée, mais en prenant en compte les interactions entre les variables d'entrée du système.

La structure de ce dernier se représente, pour notre cas d'application, de la façon suivante :

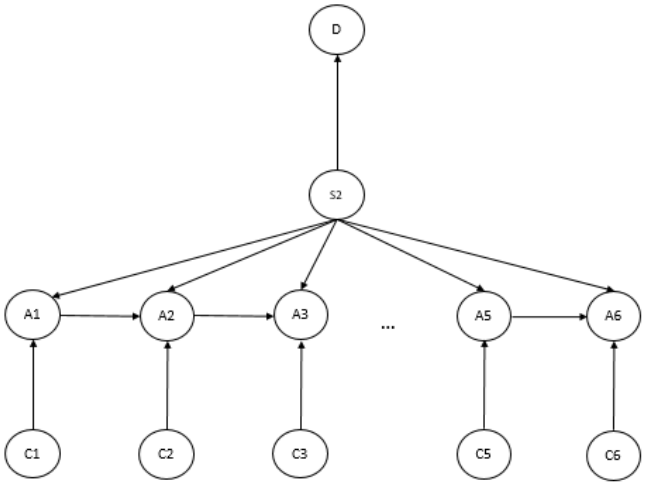


Figure 4-3: Réseau Bayésien Naïf Augmenté I

Les interactions entre les variables représentant les notes moyennes par session sont modélisées, ainsi que les interactions entre le nombre de crédits par session et les moyennes par sessions.

Comme nous connaissons le processus, en tant qu'experts, nous considérons que la valeur de la variable Ai-1 reçue à la session i-1 influence la valeur reçue à la session Ai (enchaînement dans le temps). Nous pourrions également prendre en compte l'influence des sessions d'encore avant, mais nous nous limiterons à ce modèle, pour limiter la malédiction de la dimensionnalité.

De plus, les arcs entre les variables de crédit et la variable de sortie ont été supprimés. A la place, nous modélisons seulement l'influence des crédits sur les notes moyennes aux sessions correspondantes. Nous les supprimons car nous connaissons les valeurs des variables A1, …, A6. Dans un réseau bayésien, si les arcs provenant de S2 et de Ci pointent tous vers Ai, alors, le fait de connaitre Ai ne « bloque » pas l'influence de la variable Ci sur la sortie.

La décomposition du réseau s'écrit :

$$P (D \wedge A1 \wedge C1 \wedge A2 \wedge C2 \wedge \dots \wedge A6 \wedge C6 \wedge S2) =$$

$$P(S2) \times P(D \mid S2) \times P(A1 \mid S2 \wedge C1) \times \prod_{i=2}^{6}[P(Ci) * P(Ai|S2\wedge Ci\wedge A(i-1))]$$

Nous voyons que les tables de probabilités des variables Ai (i > 1) comportent désormais trois variables parents, du fait des arcs pointant sur ces variables.

Le modèle comporte donc plus de paramètres que le modèle Naïf :

$Card(S2) + Card(D) \times Card(S2) + Card(A1) \times Card(S2) \times Card(C1) + Card(C1) + \prod_{i=2}^{6}[Card(Ai) * Card(S2) * Card(Ci) * Card\big(A(i-1)\big) + Card(Ci)]$

Enfin, nous proposons de prendre en compte l'influence du département sur les notes moyennes reçues par session. En effet, il est fort probable que les notes obtenues diffèrent selon le département en question. Nous appellerons ce réseau le réseau bayésien naïf augmenté II.

Nous inversons également le sens de l'arc entre S2 et D, de manière à ce que D devienne parent de S2.

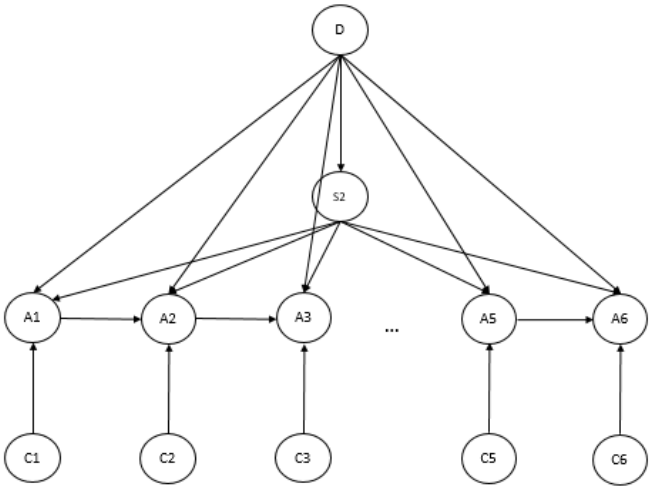


Figure 4-4: Réseau Bayésien Naïf Augmenté II

La variable liée au département pointe vers toutes les variables sur lesquelles elle est susceptible d'influer.

La décomposition du réseau s'écrit alors :

$$P\ (D \wedge A1 \wedge C1 \wedge A2 \wedge C2 \wedge \ldots \wedge A6 \wedge C6 \wedge S2) =$$

$$P(S2|D) \times P(D) \times P(A1 \mid S2 \wedge C1 \wedge D) \times \prod_{i=2}^{6}[P(Ci) * P(Ai|S2\hat{}Ci\hat{}\ A(i-1)\hat{}D)]$$

Et le nombre de paramètres augmente encore:

$$Card(S2) \times Card(D) + Card(D) + Card(A1) \times Card(S2) \times Card(C1) \times Card(D) + Card(C1) + \prod_{i=2}^{6}[Card(Ai) * Card(S2) * Card(Ci) * Card\big(A(i-1)\big) * Card(D) + Card(Ci)]$$

Le fait d'avoir la variable D comme variable parent de S2 et des variables Ai permet d'avoir des tables de probabilités qui dépendent du département. Ainsi, les modèles appris pourront être différents pour chaque département.

Dans le cadre de ce travail, nous allons appliquer deux modèles : le réseau bayésien naïf, et le réseau bayésien naïf augmenté II. Les résultats seront comparés.

4.2.2 Apprentissage des tables de probabilité

Nous devons maintenant procéder à l'apprentissage de la partie quantitative du modèle.

4.2.2.1 Caractérisation et Discrétisation des Variables

Comme nous l'avons vu précédemment, les variables sont de différents types. La variable D est catégorique, entre 1 et 11, les variables Ai sont entières, entre 0 et 4, de même que la variable S2. Les variables Ci sont des nombres entiers, allant de 0 à 18.

Les intervalles des variables étant différents, nous avons en premier lieu normalisé les variables Ai, S2 et Ci, de manière à ce que l'échelle soit la même. Nous n'avons pas normalisé D car il s'agit d'une variable catégorique.

La détermination des intervalles se fait en testant différentes valeurs, et en observant l'effet sur la qualité de la prédiction. Nous ne détaillerons pas les tests effectués dans le mémoire, mais les conclusions sont les suivantes.

Pour les variables Ai, une valeur trop faible ne permettra pas de distinguer les différentes formes dans les données en entrée. Par exemple, si le nombre d'intervalles est 2, l'outil considérera qu'un étudiant ayant obtenu une note moyenne de 4 à la session i sera équivalent à un étudiant ayant obtenu une note de 2.5 à cette même session. Une valeur plus élevée permettra de faire la différence. Cependant, un nombre trop important d'intervalles causera des difficultés liées à la malédiction de la dimensionnalité, ce qui requerra d'une part une quantité très importante de données pour combler les intervalles, et d'autre part des calculs d'inférence nettement plus couteux en termes de puissance. Nous ferons varier ce paramètre entre 5 et 10, et la choisirons valeur donnant les meilleurs résultats.

Pour les variables Ci, l'expérience a montré qu'un nombre d'intervalles ayant une valeur de 6 était optimale.

La variable D n'est pas discrétisée car elle est catégorique.

En ce qui concerne la variable S2, la discrétisation influe non seulement sur l'apprentissage et les calculs, mais également sur la résolution de la prédiction probabiliste en sortie. Ce paramètre fera partie du plan d'expérience, présenté en 3.3.

Enfin, il est important de préciser que la discrétisation des variables continues du réseau bayésien introduit la perte de l'ordonnancement des variables : les variables numériques continues deviennent des variables catégoriques.

4.2.2.2 Apprentissage des tables depuis la base de données

Dans notre cas, nous considérons que nous n'avons aucune connaissance a priori sur la forme des distributions de probabilités des différentes variables.

Nous avons donc choisi d'utiliser un apprentissage non paramétrique, en utilisant une estimation bayésienne des paramètres de la table de probabilités basée sur la formule de Laplace.

Ainsi, toutes les tables correspondantes aux termes de la distribution conjointe décomposée sont apprises depuis la base de données par estimation bayésienne avec la formule de Laplace.

4.2.3 Inférence bayésienne et Instanciation

4.2.3.1 Inférence bayésienne

Le système peut maintenant procéder à l'inférence, pour déterminer l'estimation de la variable d'intérêt S2.

Sachant la distribution conjointe P(A1^ C1^ …, C6^ D^ S2), la distribution a posteriori P(S2 | A1 ^ C1 ^ … ^ C6 ^ D) se calcule, selon l'inférence bayésienne, de la manière suivante :

$$P(S2 \mid A1 \wedge C1 \wedge \ldots \wedge C6 \wedge D) = \frac{P(A1 \wedge C1 \wedge \ldots \wedge C6 \wedge D \wedge S2)}{\sum_{S2} P(A1 \wedge C1 \wedge \ldots \wedge C6 \wedge D \wedge S2)}$$

Le calcul d'inférence est simplifié par la décomposition de la distribution conjointe, selon le réseau naïf ou naïf augmenté. Les calculs sont effectués de manière automatique par le moteur d'inférence de ProBT.

Un point doit être soulevé. Etant donné que la variable de sortie S2 est une variable continue discrétisée, l'API de ProBT considère que l'inférence bayésienne exacte ne peut pas être effectuée, car, le nombre d'intervalles peut, être (potentiellement) très grand.

Bien que dans notre cas, ce nombre n'excède pas 15, le logiciel exige l'utilisation de l'intégration de Monte-Carlo, pour fournir une estimation de la constante de marginalisation plutôt sa valeur exacte.

Cette méthode consiste à effectuer un échantillonnage, c'est-à-dire tirer un certain nombre de points depuis les termes de la distribution de probabilité décomposée, de manière à fournir une estimation moyenne de l'intégrale. Plus le nombre de points est élevé, plus l'estimation de l'intégrale se rapproche de la valeur réelle.

Dans notre cas, ProBT demande de fournir un seuil e, qui définit le critère de convergence de l'estimation de l'intégrale. Le moteur va ainsi générer des points pour l'échantillonnage jusqu'à ce que le seuil soit atteint, c'est-à-dire que l'estimation ait convergée, à la valeur du seuil près.

Nous avons choisi un seuil de 0. 01. Notre problème étant relativement simple (une seule variable à estimer, et un nombre relativement faible d'intervalles), la convergence se fait rapidement : peu de points sont requis pour avoir une bonne estimation.

Le temps de calcul pour l'apprentissage et l'inférence dépend du nombre de données et de la discrétisation, mais n'a jamais dépassé les 5 secondes lors des tests.

4.2.3.2 Instanciation

Le système instancie les valeurs des variables connues (A1=a1, C1=c1, …, C6=c6, D=d) à partir de la base de données contenant l'information sur les processus en cours, et retourne la distribution a posteriori (sous forme de fonction de masse) P (S2 | A1= a1 ^ C1=c1 ^ … ^ C6=c6^ D=d) correspondante. Cette distribution a posteriori correspond à la prédiction.

Le temps de calcul pour l'instanciation est d'environ 0.1 secondes par instance.

4.2.4 Visualisation de la prédiction probabiliste

La prédiction sur la variable de sortie S2 est présentée sous forme d'Histogramme, comme le montrent les exemples ci-dessous.

Les intervalles sont représentés en ordonnées : les différentes valeurs prises par S2. En abscisse correspond la probabilité associée.

Le paramètre important de cette représentation est la taille des intervalles de S2, qui correspond à la résolution. Nous avons représenté, pour précisément les mêmes données d'entrainement et les mêmes instances en entrée, la prédiction pour une division de S2 en quatre, puis seize intervalles. Plus le nombre d'intervalles est grand, plus la prédiction est précise. En revanche, une prédiction

plus précise requerra, comme nous le verrons lors du plan d'expérience, plus de données d'entraînement, pour donner des résultats informatifs.

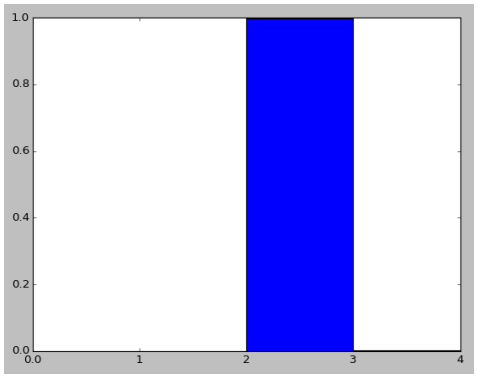


Figure 4-5: Prédiction de S2 avec un nombre d'intervalles de 4

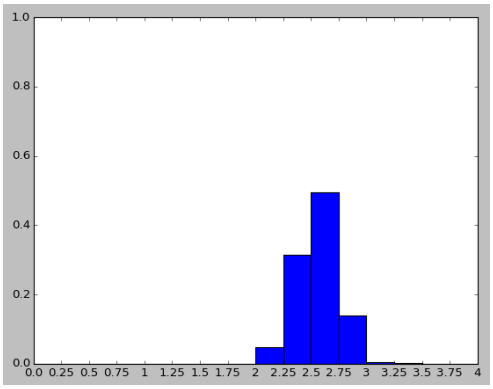


Figure 4-6: Prédiction de S2 avec un nombre d'intervalles de 16

Nous avons présenté le système, son contexte, ses conditions de fonctionnement, son architecture, ainsi que les résultats qu'il fournit.

Nous allons maintenant tester le système, pour vérifier son habilité à renvoyer des prédictions informatives, en fonction de plusieurs paramètres qui seront décrits dans la partie suivante.

CHAPITRE 5 DÉMARCHE DE TEST ET ANALYSE DES RÉSULTATS

Nous allons appliquer le système proposé, dans un premier temps sur des données simulées, de manière à tester son fonctionnement et sa robustesse, et dans un second temps sur des données réelles provenant de l'École Polytechnique, de manière à tester son fonctionnement en pratique.

5.1 Indicateur Mesuré

Pour tester le bon fonctionnement du système, c'est-à-dire la justesse des distributions de probabilités a posteriori fournies et son habilité à détecter des tendances, il nous faut définir un indicateur évaluant l'estimation probabiliste.

Nous souhaitons mesurer, pour chaque instance (étudiant) de la base de données de test, la « justesse » de l'estimation probabiliste P (S2 | C0, A0, …, A6, D) fournie par le système, par rapport à la valeur de S2 réellement obtenue pas l'étudiant. Ainsi, nous ne nous intéresserons pas ici au taux de reconnaissance, qui n'est pas un indicateur probabiliste.

L'un des indicateurs les plus utilisés dans la littérature pour tester le fonctionnement des systèmes de classification probabiliste est la fonction de perte par entropie croisée, ou « log-loss » (LL).

Il s'agit d'un indicateur intimement lié à la théorie de l'information. Il se définit de la manière suivante (toutes les distributions de probabilité sont discrètes) :

$$LLn = H(Pn, Qn) = H(Pn) + Dkl(Pn \mid Qn) = Dkl(Pn \mid Qn)$$

- Pn représente la distribution de probabilités correspondant à un histogramme concentré uniquement sur l'intervalle contenant la valeur de S2 réellement obtenue par l'étudiant n
- Qn représente la distribution de probabilités fournie par le système pour l'étudiant n, soit P (S2 | C0=c0, A0=a0, …, A6=a6, D=d) dans notre cas.
- H(Pn, Qn) est l'entropie croisée entre Pn et Qn
- H(Pn) est l'entropie de Pn, qui est nulle
- Dkl(Pn | Qn) est la divergence de Kullback-Leibler entre la distribution «réelle » Pn et la distribution estimée Qn.

La divergence de Kullback-Leibler entre Pn et Qn se définit de la manière suivante :

$$\text{Dkl (Pn |Qn)} = \sum_i Pn(i) log \frac{Pn(i)}{Qn(i)} = -\sum_i Pn(i) \log Qn(i)$$

Où i correspond à l'intervalle i de la distribution considérée.

L'indicateur est toujours positif.

Le log-loss peut aussi être vu comme l'inverse du log-vraissemblance des vériatables classes sachant les distributions de probabilités prédites.

Nous pouvons voir que le log-loss va punir fortement les estimations probabilistes ayant faiblement estimé la probabilité de l'intervalle i alors que ce même intervalle contient la valeur réelle de S2 pour l'étudiant n considéré. Elle aura une valeur infinie si l'estimation est nulle. Cependant, grâce à l'estimation bayésienne des paramètres, les probabilités ne sont jamais tout à fait nulles. Nous avons tout de même défini une valeur maximale, fixée à 100.

Inversement, il tendra vers 0 si l'estimation probabiliste corresond à un unique batonnet centré sur l'intervelle contenant la valeur réelle de S2.

A des fins d'illustration, nous présentons un exemple de calcul d'erreur log-loss représentatif de ce qui est effectué lors des tests de ce mémoire.

Dans notre exemple, la variable de sortie S2 est discrétisée en 8 intervalles.

Supposons que l'étudiant n testé finisse avec une valeur de S2 de 3.1 (figure 5.1). Cela rentre dans l'intervalle [3,3.5[. Ainsi, la distribution Pn correspondante est représentée figure 5.2.

D'autre part, le modèle fournit une prédiction probabiliste Qn, représentée en figure 5.3.

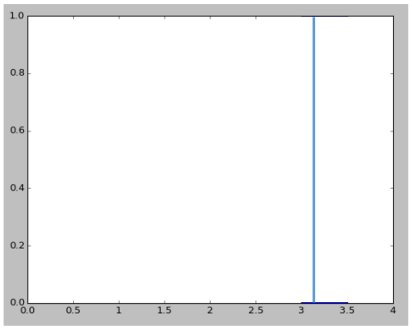


Figure 5-1: Valeur S2 réelle de l'étudiant n

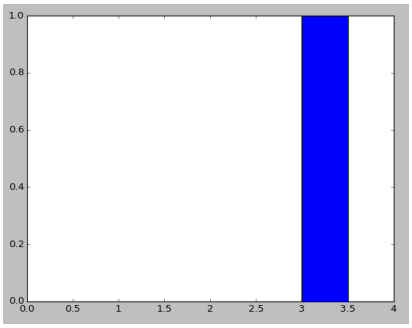


Figure 5-2: Distribution Pn

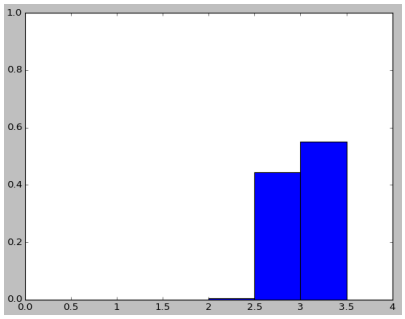


Figure 5-3: Distribution prédite Qn

Ainsi, le log-loss pour le cas de l'étudiant n correspond à la divergence de Kullback Leibler entre Pn et Qn. Cette valeur serait nulle si Qn était exactement égale à Pn. Elle serait infinie si la probabilité de l'intervalle i=[3,3.5[de Qn était nulle.

Dans le cas présenté en figure 5.3, le log-loss a pour valeur 0.63.

D'autre part, si Qn (distribution prédite) était une distribution uniforme (figure 5.4), le log-loss aurait pour valeur 2.07.

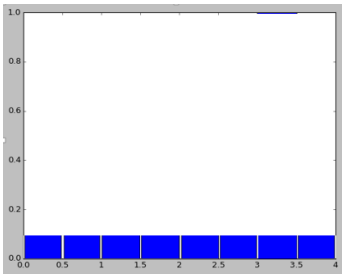


Figure 5-4: Distribution uniforme

Physiquement, le log-loss représente la quantité d'information (en bits) que l'on doit ajouter à l'estimation probabiliste Qn fournie par le système pour pouvoir connaitre exactement l'intervalle i contenant la valeur S2 obtenue par l'étudiant n (c'est-à-dire pour pouvoir connaitre Pn).

Nous allons nous intéresser plus généralement au log-loss moyen, qui se définit par :

$$\text{LLmoy} = \frac{1}{N}\sum_{n=1}^{N} LLn = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i} Pn(i)\log Qn(i)$$

Où N correspond au nombre d'étudiants considérés lors du test (nombre de données de test).

Lors des tests, nous comparerons les log-loss moyens obtenus avec le log-loss obtenu si l'estimation fournie par le modèle est une distribution uniforme, soit le cas le moins informatif, où toutes les probabilités sont égales, que nous dénomerons LL(Loi Uniforme).

Lors des simulations, nous comparerons les log-loss moyens obtenus avec le log-loss des distributions a posteriori réelles, pour avoir un idée de la performance du système. Ces dernières ne sont pas connues avec les données réelles, c'est pour cela que nous ne pourrons pas les utiliser dans le test avec les données réelles.

D'autre part, il est important de préciser que LL(Loi Uniforme) et LLmoy(Système) dépendent de la discrétisation de la variable de sortie S2, d'où le besoin de considérer un indicateur permettant la comparaison des résultats en fonction de cette discrétisation.

Nous proposons donc, dans le cadre de ce travail, de nous intéresser à un indicateur que nous allons dénommer « Gain d'information » obtenu avec notre système. Celui-ci se définit par rapport au log-loss de la loi uniforme, pour une discrétisation donnée :

Gain = LL (Loi Uniforme) – LLmoy (Système)

Ce gain mesure la quantité d'information apportée par le système par rapport à l'information fournie par la loi uniforme pour estimer l'intervalle dans lequel se situe la valeur de S2. Ainsi, plus sa valeur est grande, plus le système apporte de l'information (en moyenne) à l'utilisateur pour savoir précisément l'intervalle dans lequel se trouve la moyenne finale des étudiants.

Nous allons donc comparer les gains obtenus avec différentes discrétisations, dans le but de déterminer quelle discrétisation convient le mieux en fonction des données disponibles

L'unité de ces indicateurs est le bit (relatif à la quantité d'information).

De manière à illustrer ces propos, nous présentons un exemple ci-dessous, pour une seule donnée de test, dont la valeur réelle de S2 est de 3.25. Supposons que S2 soit discrétisée en 2 intervalles. Supposons que la prédiction fournie par le système soit idéale, et retourne une erreur log-loss de 0 (figure 5.5 A).

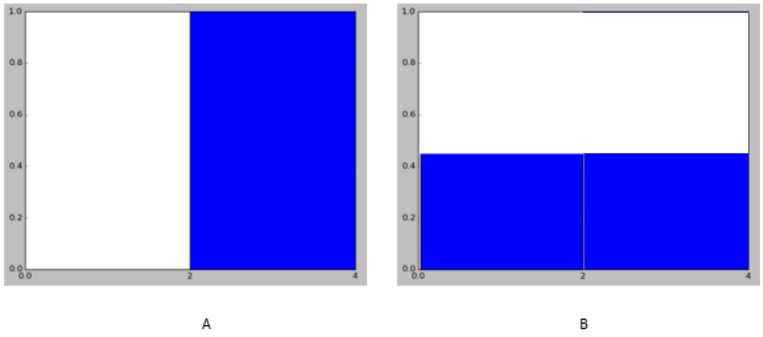


Figure 5-5 : Sortie idéale (A) et uniforme (B) pour une dicrétisation en 2 intervalles

D'autre part, le log-loss de la loi uniforme, si S2 est discrétisée en 2 intervalles (figure 5.5 B) a pour valeur 0.7.Ainsi, le gain apporté par le système si la discrétisation a pour valeur 2 sera : 0.7-0 =0.7.

Si la discrétisation se fait en 8 intervalles, (figure 5.6 A), nous obtenons une erreur log-loss de 0 également si la prédiction est idéale. Or, la loi uniforme a pour log-loss 2.07 (figure 5.6 B). Ainsi, le gain d'information sera de 2.07-0= 2.07 bits.

Nous voyons donc que les erreur log-loss ne suffisent pas à comparer les résultats obtenus si la discrétisation diffère. Ainsi, nous proposons le gain pour choisir la discrétition qui maximise l'apport en information. Dans l'exemple présenté, le choix du nombre d'intervalles de 8 s'impose, car le gain est plus important. Cependant, comme nous le verrons au cours de l'expérience, ce choix dépendera du nombre de données disponible (une discrétisation trop importante avec un nombre de données trop faible donnera un gain faible).

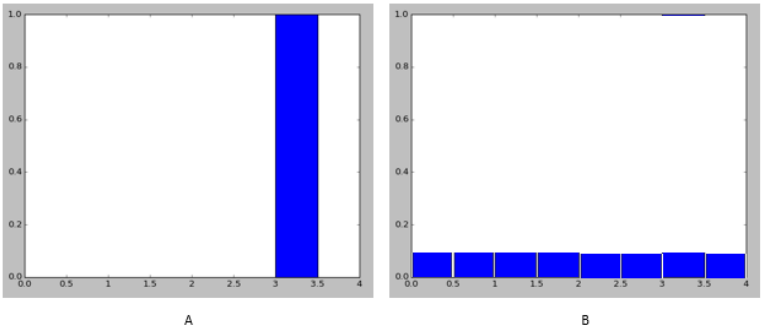


Figure 5-6 : sortie idéale (A) et uniforme (B) pour une discrétisation en 8 intervalles

5.2 Expérience en données simulées

5.2.1 Génération des Données

Avant d'appliquer le système sur les données réelles, il nous est paru important de l'appliquer sur des données générées, de manière à bien comprendre le système et ses limites.

Le but étant d'évaluer en fonction du nombre de données d'entrainement disponibles, du bruit dans les données, de l'incertitude, de la précision voulue et du type de réseau choisi (Naïf ou Naïf Augmenté) :

- L'habilité du système à détecter des tendances dans les données d'entrée
- L'estimation probabiliste fournie par le système

L'idée de la simulation est de générer des familles de tendances, ou trajectoires, d'étudiants, évoluant durant leur baccalauréat, de la première à la douzième session.

Lors de cette simulation, nous respectons les conditions de fonctionnement du système, en particulier l'hypothèse de distribution identique et uniforme des exemples.

La Figure 5.7 présente un exemple de famille de tendances à simuler. En ordonnée est représentée la note moyenne obtenue à la session correspondante en abscisse. Il s'agit d'une famille de tendance, dans le sens où plusieurs tendances uniques (étudiants fictifs) constituent cette famille.

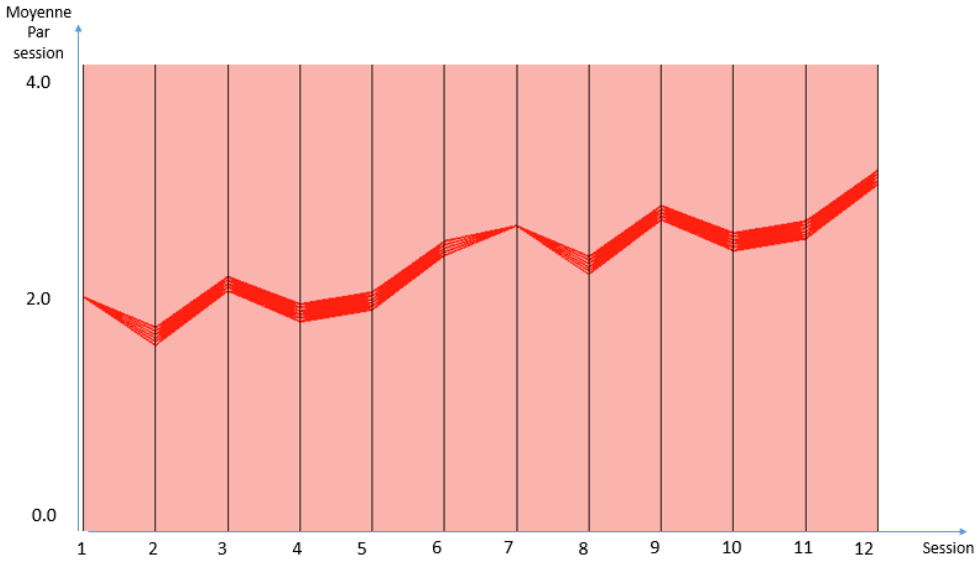


Figure 5-7 : exemple de famille de tendances à simuler

Dans le cadre de notre simulation, nous choisissons de fixer le département.

Comme nous l'avons vu, le système ici proposé prend seulement en compte les notes obtenues et les crédits pris jusqu'à la session 6, et considère la moyenne cumulative obtenue lors des sessions suivantes comme la variable à prédire.

Ainsi, lors de la simulation, nous ne générerons pas les tendances en entier, mais seulement sur les six premières sessions, et nous associerons à chaque élément de la famille de tendance considérée une valeur unique sur la sortie S2, prise depuis une distribution de probabilité qui est propre à la famille.

Ainsi, à chaque famille de tendance correspond une distribution de probabilités sur S2 qui lui est propre.

La figure 5.8 illustre, pour la même famille de tendance que plus haut, la manière dont les tendances ont été générées. Nous utilisons les coordonnées parallèles pour visualiser la famille.

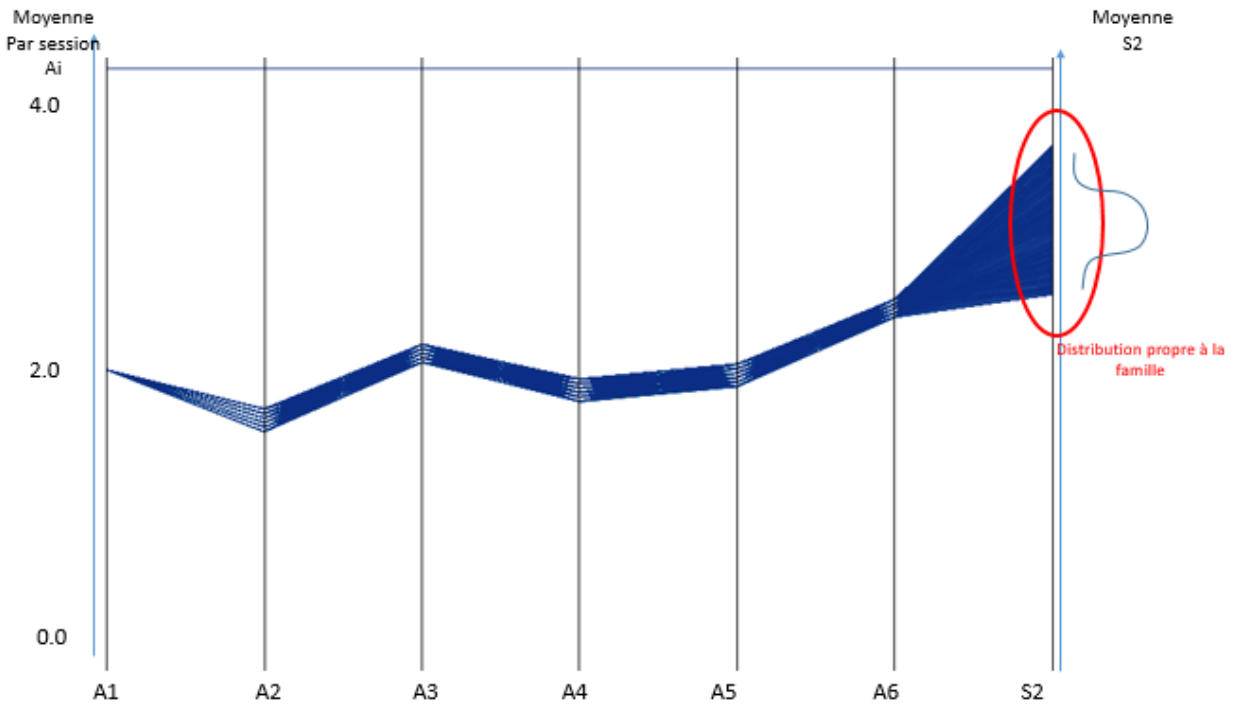


Figure 5-8 : Visualisation en coordonnées parallèles de la famille simulée

Nous avons choisi de simuler les familles de tendance de cette manière dans le but de simuler l'incertitude. En effet, sur l'exemple ci-dessus, nous pouvons voir que le fait de connaitre à quelle famille de tendance appartient un étudiant n'est pas suffisant pour connaitre exactement la moyenne (S2) qu'il aura lors de la seconde partie de son baccalauréat, puisque S2 est répartie selon une distribution de probabilités, et non pas une valeur unique (qui serait une distribution dite Dirac).

Il y a ainsi plusieurs solutions possibles pour S2, et le système sera chargé de fournir une estimation probabiliste de cette variable, se rapprochant de la distribution « réelle » sur S2 correspondant à la famille de tendances considérée.

La simulation génère 16 familles de tendances, générées à partir de courbes représentées sur la figure 5.9. Il est important de préciser que le but de la génération de données est moins de simuler fidèlement tous les comportements possibles que de tester l'habilité du système à détecter les tendances, et à estimer la sortie correspondante de manière probabiliste.

Ainsi, nous nous limitons à 16 familles, dont 8 partent d'une moyenne de 2 pour la première session, et 8 partent d'une moyenne de 4. Ainsi, nous couvrons à 0.5 près toutes les valeurs possibles à la 6ème session.

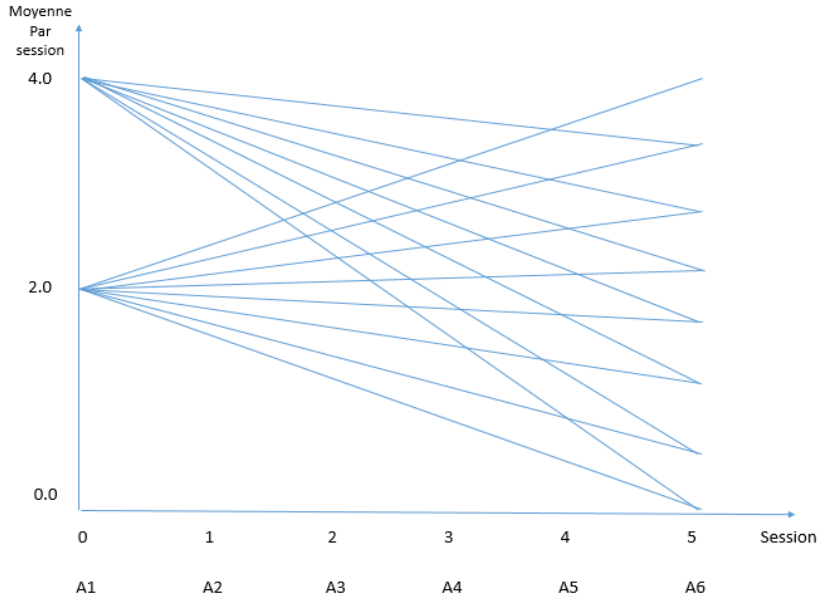


Figure 5-9 : courbes définissant les différentes familles de tendances

Les données sont générées de la manière suivante :

Chaque famille de tendances « a » est générée séparément. Pour chaque famille, on génère des données pour chaque variable Ai, sous forme de vecteur de taille m, selon l'équation suivante :

- **Ai** = **Ai-1** + f (a, i, origine) **U** – 0.03 **Ci** + b **U**

- **Ai-1** est le vecteur de la session précédente, de taille m, où m est le nombre d'exemples par famille.
- f (a, i, origine) est une fonction qui dépend de a (: la famille de tendance concernée) et de i (: la session i). Ainsi, on génère une évolution de Ai en fonction de i, propre à chaque famille. Les différentes fonctions F(a) sont représentées ci-dessus.
- « Origine » correspond à l'ordonnée à l'origine de la droite.
- **U** est le vecteur de taille m, dont toutes les valeurs sont égales à 1.
- **Ci** est le vecteur de taille m contenant le nombre de crédits, pris par chacun des m étudiants à la session i. Plus le nombre est élevé, plus la moyenne générée pour la session i baisse. Ce paramètre est responsable de la déformation de la famille présentée dans la figure 5.8.

- « b » correspond au terme de bruit, qui déforme la tendance (qui n'a rien à voir avec le nombre de crédits), qui est tiré selon la loi Normale de moyenne 0 et d'écart type Sigma_Bruit.

A chaque famille « a » on associe un vecteur S2, aussi de taille m. Ce vecteur est généré de la manière suivante, avec la librairie Numpy :

S2 (a) = Distrib (a) x **U**

Où Distrib correspond à une distribution de probabilités a posteriori, propre à la famille « a », qui simule l'incertitude. Ainsi, le vecteur S2(a),de taille m, comporte m éléments qui sont tirés aléatoirement et indépendamment depuis la distribution Distrib(a).

Au cours du plan d'expérience, nous considérerons 3 cas :

- Distrib (a) = Dirac (h) : La distribution est un Dirac (aucune incertitude), centré sur la valeur h. La valeur de h, propre à a, est détaillée en annexe A.
- Distrib (a) = Normale (h, 0.25) : La distribution est une loi Normale, centrée sur la valeur h, d'écart type 0.25, pour simuler de l'incertitude.
- Distrib (a) = Uniforme ([0,4]) : La distribution est une loi Uniforme entre 0 et 4. Il s'agit d'un cas d'incertitude critique, c'est-à-dire que la connaissance de la tendance d'entrée n'explique en rien la variation en sortie.

Plus de détails sur la simulation sont fournis en annexe A. En particulier une illustration des différents paramètres de la simulation pour une famille donnée.

Une fois les 16 familles générées séparément, on les rassemble dans un fichier csv, qui constitue la base de données d'apprentissage. La taille m des vecteurs sera l'un des paramètres du test.

En parallèle, on génère de nouveau les 16 mêmes familles, avec les mêmes paramètres, mais avec un nombre m fixé à 60. Nous les rassemblons dans un même fichier csv que nous considérerons comme la base de données de test. Nous mesurerons les log-loss moyens et gains obtenus sur ce fichier de test.

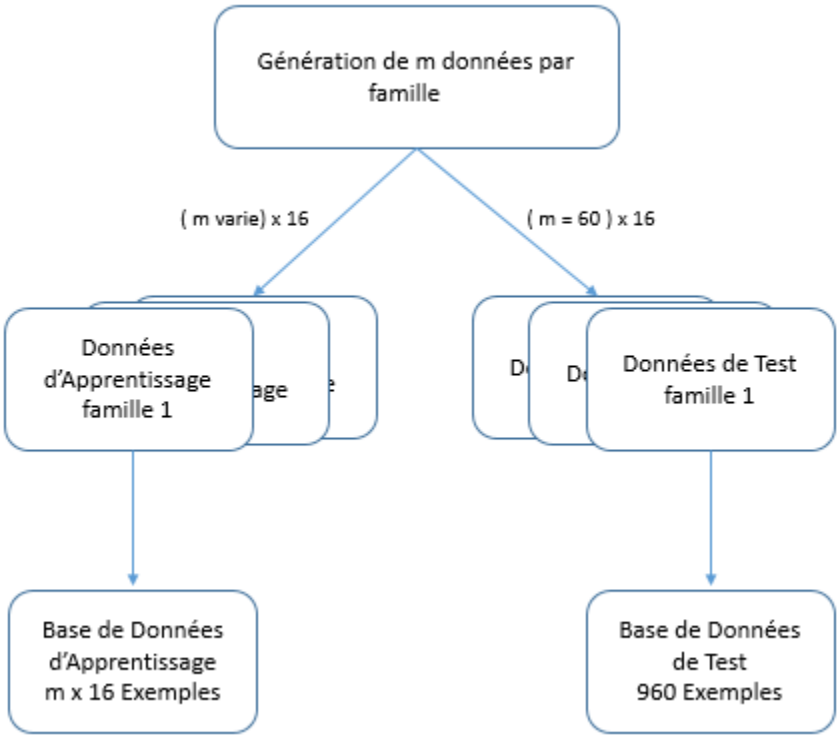


Figure 5-10 : Illustration de la procédure de génération de données

5.2.2 Plan d'expérience

Nous faisons varier plusieurs paramètres. Certains sont propres au système, d'autres sont relatifs aux données générées.

Le tableau ci-dessous résume les paramètres importants.

Table 5-1 : Paramètres de l'Expérience en Données Simulées

Indicateur Mesuré	Paramètres de la Simulation			Paramètres du Système	
	Incertitude sur S2	Nombre de données m	Bruit	Discretisation de S2	Forme du Réseau
Log Loss	Dirac, Loi Normale, Loi Uniforme	5, 20, 200	Sigma= 0, 0.15, 1	4, 8, 15	Naif, Naif Augmenté

Parmi les paramètres propres au système, nous nous intéresserons :

- A la discrétisation de la variable de sortie S2, qui correspond à la résolution, visible par l'utilisateur, sur la prédiction. Nous nous attendons à ce que le système ait plus de mal à ressortir une estimation ayant une erreur faible sur un nombre d'intervalles élevé (15) que sur un nombre d'intervalles faible (4) si le nombre de données d'apprentissage est faible. En revanche, si le nombre de données est suffisamment élevé, la prédiction obtenue avec un nombre d'intervalles élevé devrait être plus informative que celle obtenue avec un nombre d'intervalles faible. Nous utiliserons le gain d'information pour comparer les performances.
- A la structure du réseau. Nous nous attendons à ce que le réseau naïf fournisse des résultats moins performants que le réseau naïf augmenté, compte tenu du biais du modèle.

En ce qui concerne les paramètres propres aux données générées :

- L'incertitude sur la sortie S2. Comme nous l'avons vu, nous la simulons en générant une distribution de probabilités qui est propre à une famille de tendance donnée. Nous ferons varier cette incertitude, d'un état certain (Distribution en Dirac, propre à la famille de tendances en question), vers un état plus incertain (distribution normale, propre à la famille de tendances en question), et enfin, vers un état d'incertitude critique, où toutes les familles de tendances mènent vers une même distribution uniforme répartie entre 0 et 4.
- Le nombre n de données d'entrainement par famille de tendances : 5, puis 20, puis 200. Il est très important de comprendre qu'il s'agit du nombre de données par famille de tendances. Ainsi, le nombre total de données d'entrainement s'élèvera à 16n.
- Le bruit dans les tendances en entrée : nous allons progressivement déformer les tendances en entrée, avec le terme de bruit, de manière à rendre la détection de tendance plus complexe. Le cas critique est également présent, avec une valeur de Sigma à 1.0. Si ce cas sort des conditions de fonctionnement du système, nous avons voulu tester le comportement du système dans des conditions critiques.

Nous avons fixé deux paramètres :

- Tout d'abord en ce qui concerne les données, le département. Nous avons considéré que le but de la simulation est de tester l'habilité du système à détecter des tendances, pour un département donné. Comme nous l'avons vu, il est possible d'apprendre des modèles spécifiques pour chaque département, grâce à la variable D. Ainsi, nous avons choisi de ne pas modéliser des comportements différents selon les départements, étant donné que pour chaque département D, l'enjeu sera le même : apprendre les tendances, et les reconnaitre.
- Ensuite, par rapport au système, la discrétisation des variables d'entrée. Nous avons fixé la discrétisation de Ai à 10, et celle de Ci à 5, car l'expérience a montré que ces valeurs sont optimales pour détecter les tendances avec les données générées.

Les hypothèses à vérifier sont les suivantes. Nous les testerons en fonction des paramètres du plan d'expérience.

H0a : Lorsqu'aucun bruit n'est présent dans les tendances d'entrées, le système fournit des estimations probabilistes (histogrammes) qui ont une erreur log-loss moyenne minimale (qui correspond au log-loss de la distribution a posteriori réelle).

H0b : Le système est robuste face au bruit des tendances d'entrée : le log-loss moyen n'augmente pas de manière significative si le bruit augmente.

H0c : Les log-loss moyens obtenus avec le réseau naïf sont plus élevés que ceux obtenu avec le réseau naïf augmenté.

H0d : Augmenter le discrétisation de la variable de sortie S2 permet d'augmenter le gain d'information moyen apporté par le système.

5.3 Analyse de l'expérience en données simulées

Le plan d'expérience décrit pour l'expérience en question a été suivi. Les résultats obtenus ont été reportés en Annexe B.

Le temps de calcul pour l'apprentissage et l'inférence dépend du nombre d'instances en apprentissage et de la discrétisation, mais varie globalement de 0.5 (5 exemples par famille) à 5 secondes (200 exemples par famille).

Le temps de calcul pour l'interrogation du réseau est d'environ 0.1 seconde pour une instance de test (CPU Quad-Core, 2.58 GB).

Intéressons-nous tout d'abord à l'évolution du log-loss moyen en fonction du nombre de données.

5.3.1 Effet du nombre de données, du bruit et de l'incertitude en sortie

Dans un premier temps, intéressons-nous seulement à l'effet des paramètres propres aux données, à savoir le nombre de données, le bruit déformant les tendances et la forme de la distribution en sortie. Nous ne considérons donc pas l'effet de la discrétisation de S2 dans cette partie.

Les graphiques ci-dessous représentent l'évolution du log-loss moyen en fonction du nombre de données n pour un niveau de bruit, une distribution en sortie, et un réseau bayésien donné. Le nombre d'intervalles est fixé à 8. Cependant, les résultats sont les mêmes pour un nombre de 4 ou de 15 (voir annexe B).

Intéressons-nous à l'évolution de l'erreur si la distribution de sortie est un Dirac :

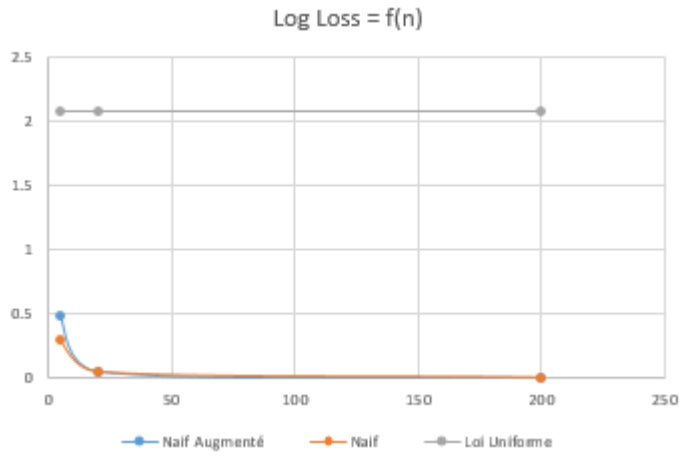


Figure 5-11 : Evolution du log Loss en fonction de n pour Sigma_Bruit = 0

Figure 5-12 : Evolution du log Loss en fonction de n pour Sigma_Bruit = 0.15

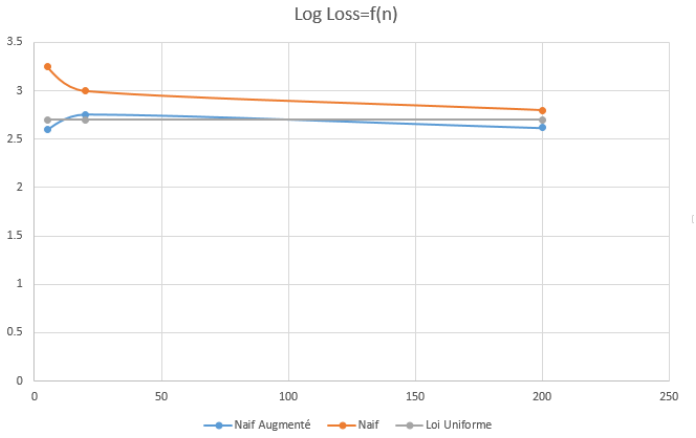


Figure 5-13 : Evolution du log Loss en fonction de n pour Sigma_Bruit = 1.0

Nous voyons que, dans le cas où la sortie est un Dirac, si le bruit en entrée est nul, c'est-à-dire que les tendances ne sont pas distordues, l'erreur moyenne descend rapidement vers 0 (: la valeur du log-loss minimal dans le cas où la distribution a posteriori réelle est un Dirac) avec le nombre de données d'entrainement. Ainsi, le système parvient très bien à détecter les différentes tendances, et à estimer la sortie associée (figure 5.14).

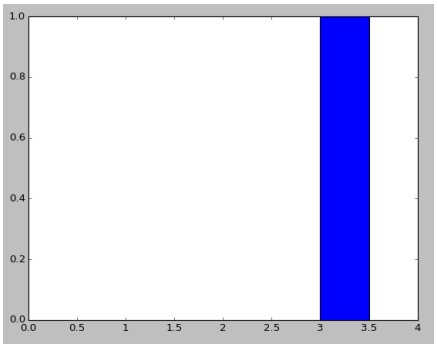


Figure 5-14 : Exemple de sortie si Sigma_Bruit =0

Lorsque le bruit augmente (Sigma = 0.15), l'erreur augmente également. Augmenter fortement le nombre de données ne permet plus de diminuer l'erreur, qui devient alors constante en fonction de n à partir de n=20. Ceci est lié au fait que les tendances se distordent, et tendent à se ressembler de plus en plus. Ainsi, il devient plus difficile pour le système de les différencier. L'histogramme prédictif fournit pour ne correspond plus à un unique bâtonnet centré sur l'intervalle contenant la valeur réelle de S2, mais à un histogramme dont la prédiction est répartie sur plusieurs intervalles (figure 5.15), correspondants aux sorties correspondant aux familles « ressemblant » à la tendance en entrée.

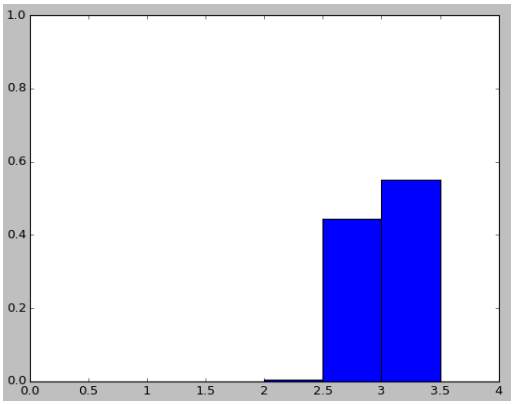


Figure 5-15: Exemple de sortie si Sigma_Bruit =0.15

Lorsque le bruit est critique (Sigma = 1.0), le système ne peut plus distinguer les tendances, et ressort alors des estimations se rapprochant de la distribution uniforme (figure 5.16). Ceci explique le fait que le log-loss moyen obtenu se rapproche de celui correspondant à l'estimation de la loi uniforme.

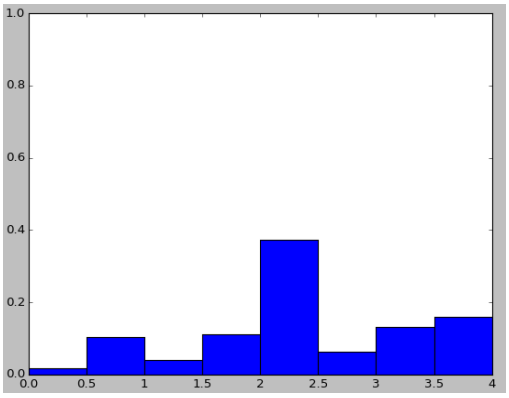


Figure 5-16: Exemple de sortie si Sigma_Bruit =1.0

Nous pouvons donc conclure que, pour une discrétisation de la sortie donnée, si la distribution de sortie est un Dirac :

- De manière générale, l'erreur diminue avec le nombre de données.
- Si aucun bruit n'est présent en entrée, le système détecte parfaitement les tendances, et ressort une estimation très juste de la sortie S2 : le log-loss est minimal, égal à 0. H0a est donc validée
- Si un bruit modéré (Sigma = 0.15) est présent en entrée, le système a plus de mal à détecter les tendances, mais l'erreur log-loss reste globalement faible par rapport à la loi uniforme. Nous validons donc l'hypothèse H0b.
- Si un bruit critique (Sigma = 1.0) est présent en entrée, le système ne détecte plus les tendances. L'estimation est totalement biaisée, et aucune information ne peut plus être tirée des données. Nous rejetons donc l'hypothèse H0b.
- Da manière générale, le log loss moyen obtenu avec le réseau naïf augmenté n'est pas plus faible que celui obtenu avec le réseau naïf. H0c est donc réfutée.

Intéressons-nous à l'évolution de l'erreur si la distribution de sortie est une Loi Normale :

Nous avons ajouté aux graphiques ci-dessous la courbe correspondant au log-loss moyen de la distribution de sortie réelle correspondant à la famille de la tendance d'entrée. Cette valeur d'erreur correspond à l'erreur log-loss minimale atteignable par le système. Dans le cas où la sortie est une loi normale (et non plus un Dirac), cette valeur n'est plus nulle, car la Loi Normale génère de l'incertitude.

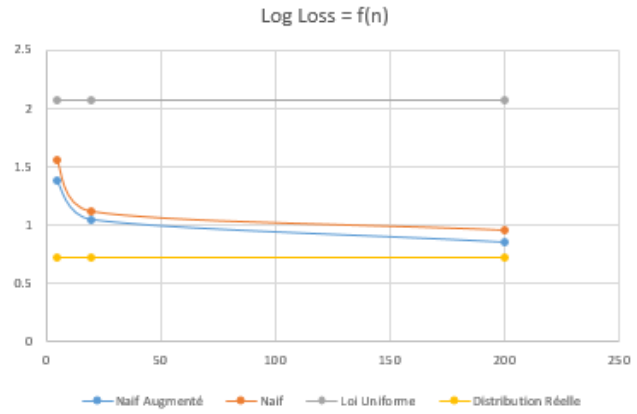


Figure 5-17 : Evolution du log Loss en fonction de n pour Sigma_Bruit = 0

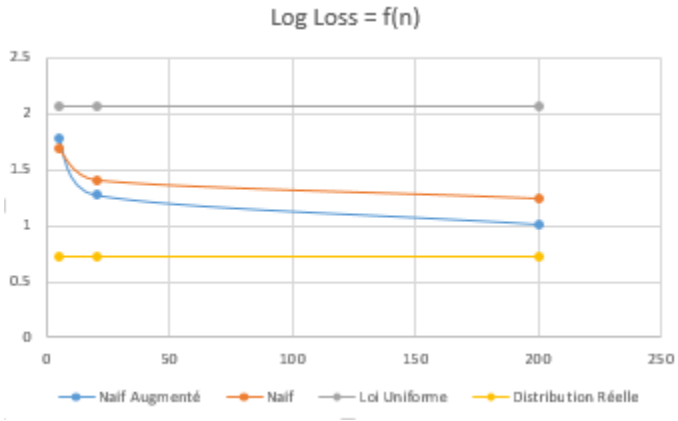


Figure 5-18 : Evolution du log Loss en fonction de n pour Sigma_Bruit = 0.15

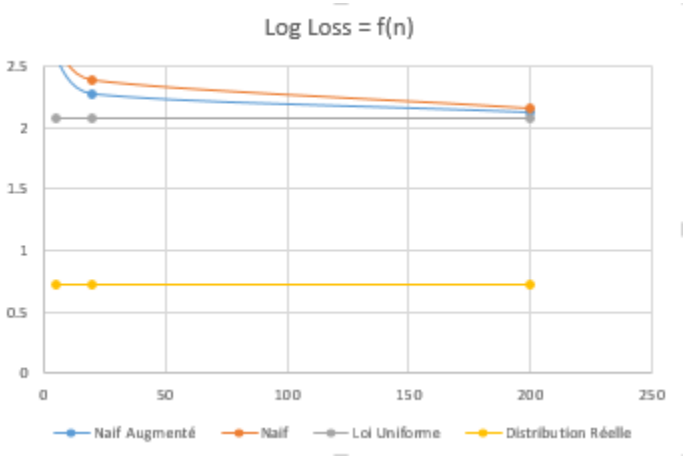


Figure 5-19 : Evolution du log Loss en fonction de n pour Sigma_Bruit = 1.0

Similairement à l'étude précédente, le rejet ou la validation de l'hypothèse H0b va dépendre du niveau de bruit, et du nombre de données d'entrainement. Des niveaux de bruit ayant Sigma_Bruit =0 et Sigma_Bruit = 0.15 conduisent à accepter H0b, tant que le nombre de données n'est pas critique (inférieur à 20 exemples par famille). Cependant, avoir un niveau de bruit critique (Sigma_Bruit=1.0) conduira toujours au rejet de l'hypothèse, car le log loss a un niveau égal ou supérieur à celui de la distribution uniforme.

D'autre part, l'erreur de l'estimation fournie par le réseau naïf augmenté est toujours plus faible que celle fournie par le réseau naïf. L'hypothèse H0c est donc validée lorsque la distribution de sortie en sortie n'est plus un Dirac. Le biais du réseau naïf est maintenant mis en évidence.

Il est ici très intéressant de voir que, contrairement au cas de l'étude précédente, même lorsque le bruit en entrée est nul, l'estimation probabiliste fournie par le système, avec les deux réseaux, n'atteint jamais la valeur d'erreur log-loss minimale. Et ce, même lorsque le nombre de données est très important. Nous devons donc rejeter l'hypothèse H0a.

Ce point permet de mettre en évidence une limite importante du système. En effet, cela signifie que la distribution probabiliste fournie (histogramme) pour un étudiant donné ne correspondra jamais exactement à la distribution a posteriori réelle correspondant à la famille correspondante. L'estimation probabiliste est donc biaisée, même lorsque le réseau naïf augmenté est utilisé (bien que le biais soit moins important qu'avec le réseau naif). Pour illustrer ce point, les figures ci-

dessous ont été représentées. Les histogrammes ont été obtenus avec un niveau de bruit nul, et un nombre d'exemples d'entrainement par famille de 200, avec le réseau bayésien naïf augmenté.

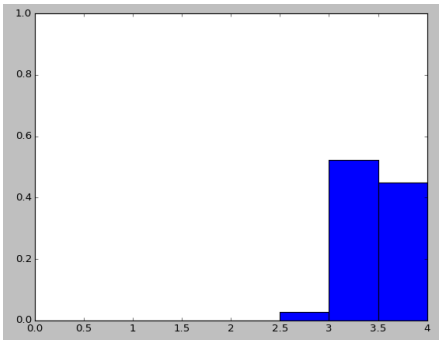


Figure 5-20 : Distribution a posteriori Réelle de S2 (prédiction idéale), discrétisée, pour tous les étudiants de la famille i

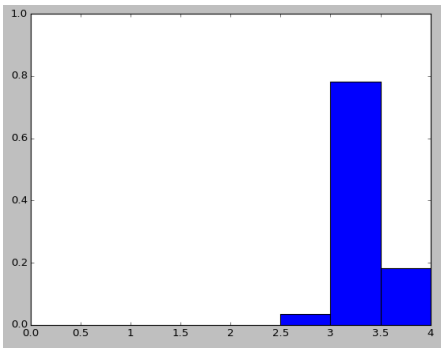


Figure 5-21 : Prédiction de S2 fournie par le système, pour un étudiant appartenant à la famille i

Nous pouvons constater que le système fournit une distribution a posteriori de S2 distordue, dans le sens où il tend à favoriser les valeurs les plus fréquentes, comme ici l'intervalle [3, 3.5]. Cette déformation est encore plus importante avec le réseau naïf. Ceci explique les log-loss moyens plus élevés que celui de la distribution a posteriori réelle.

Si la distribution de sortie est une loi Uniforme entre 0 et 4 :

Les résultats obtenus sont globalement les mêmes pour toutes les valeurs de bruit. Les évolutions suivent le graphique ci-dessous.

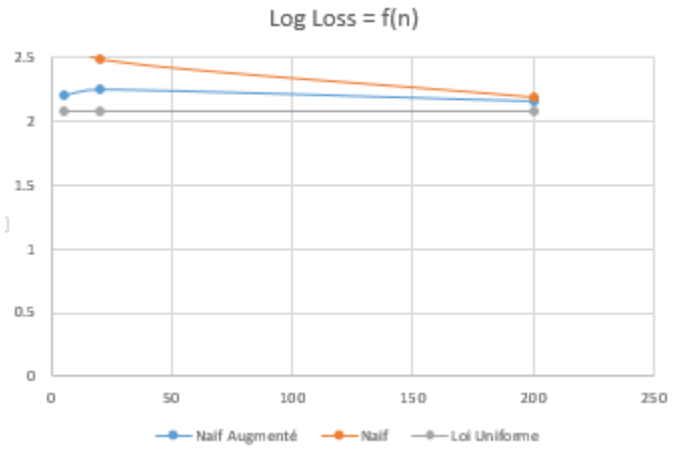


Figure 5-22 : Evolution du log Loss en fonction de n pour Sigma_Bruit = 0.15

En ce qui concerne le réseau naif augmenté, l'estimation probabiliste reste globalement proche d'une loi uniforme, peu importe la quantité de données. Cependant, le réseau naif a tendance à fournir une estimation très fausse lorsque le nombre de donnée est faible. Il peut donc fournir des prédictions trompeuses pour l'utilisateur, laissant croire que la sortie est prévisible (: distribution non uniforme) alors que la distribution réelle est une loi uniforme.

Pour conclure cette première analyse, nous pouvons dire que le système parvient bien à distinguer les tendances lorsque le bruit est absent. Lorsque la sortie est très certaine, l'estimation fournie est très juste. Lorsque le bruit augmente, le système va de plus en plus confondre les familles auxquelles appartiennent les tendances d'entrée, et ainsi fournir une prédiction se rapprochant de la loi uniforme.

Lorsque la distribution en sortie est plus complexe (Loi Normale, d'écart type 0.25), une limite important du système est mise en évidence. Il a en effet tendance à être « trop confient », en favorisant les valeurs les plus fréquentes, et en sous estimant ainsi les probabilités liées aux valeurs moins fréquentes. Ceci peut se révéler problématique, dans le sens où certaines valeurs probables de la sortie peuvent être ignorées, et ainsi mener à des décisions ignorant des risques éventuels.

Nous avons vus que le réseau naïf fournit une estimation encore plus biaisée que celle du réseau augmenté. La littérature a déjà abordé le sujet de l'estimation probabiliste fournie par le réseau naïf

(Domingos & Pazzani, 1996). Ce dernier est en effet reconnu pour fournir des estimations trop confiantes de la sortie, ignorant ainsi les autres possibilités. La principale cause de cette « discrimination » est attribuée au biais introduit par l'hypothèse d'indépendance des variables d'entrée. Ceci explique la meilleure performance du réseau naïf augmenté, qui tient compte de la dépendance entre les variables. Cependant le réseau naïf augmenté présente toujours un biais dans la prédiction probabiliste. L'estimation probabiliste de manière générale n'est pas reconnue comme une tâche simple. (Niculescu-Mizil & Caruana , 2005) ont comparé différents algorithmes en ce qui concerne leur habilité à fournir une estimation probabiliste juste, et un biais était toujours présent (plus ou moins selon l'algorithme utilisé). Des méthodes ont été proposées pour diminuer davantage les distorsions, comme la calibration de Platt ou la régression isotonique (Platt, 1999), et les résultats ont souvent été satisfaisants. Il serait ainsi intéressant, lors d'études futures, d'ajouter ces méthodes au système, de manière à vérifier si l'estimation s'améliore.

5.3.2 Effet de la discrétisation de la variable de sortie S2

Nous avons étudié, pour une valeur de discrétisation de S2 fixée (8), les effets du nombre de données d'entrainement, du bruit en entrée, de la distribution en sortie, et de la structure du réseau.

Nous allons maintenant étudier l'influence de la discrétisation de S2 (la discrétisation de la sortie), en analysant l'évolution du Gain (définit en 5.1) en fonction du nombre de données, pour le réseau naïf augmenté, en fixant le bruit en entrée (Sigma =0.15).

Si la distribution de sortie est un Dirac :

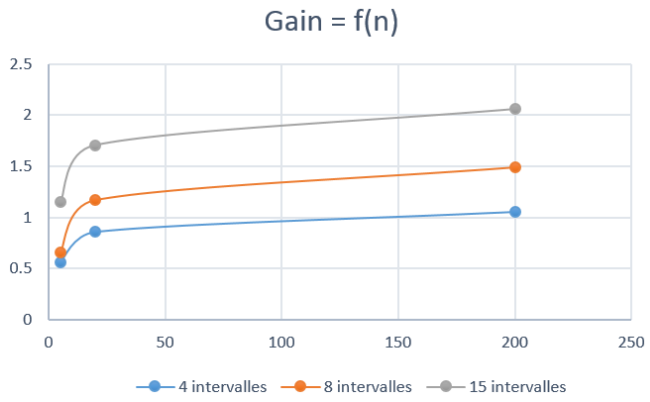


Figure 5-23 : Évolution du Gain en fonction du nombre de données d'entrainement

Nous pouvons voir que le gain augmente avec le nombre d'intervalles de discrétisation de S2, peu importe le nombre de données. Ainsi, si la distribution de sortie est un Dirac, avoir un système ayant une résolution importante est toujours plus informatif qu'avec une résolution plus faible. L'hypothèse H0d est donc toujours acceptée.

Si la distribution de sortie est une loi Normale (nous avons pris plus de points, en particulier en prenant un nombre d'exemples d'entrainement de 100, et de 2000) :

Gain :

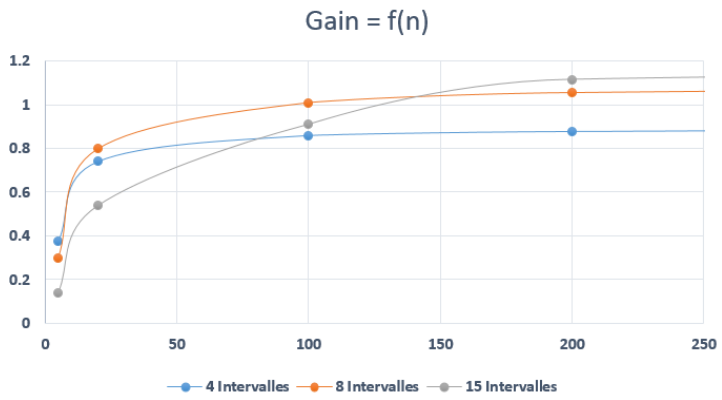


Figure 5-24 : Évolution du Gain en fonction du nombre de données d'entrainement

Dans le cas où la distribution de sortie est une loi normale, les évolutions sont plus complexes, mais très intéressantes. Nous voyons que, lorsque le nombre de données est faible, l'estimation la plus informative est celle fournie par une discrétisation à 4 intervalles. Lorsque le nombre de données augmente (n=20 exemples par famille), l'estimation optimale devient celle fournie par une discrétisation à 8 intervalles. Enfin, quand le nombre de données devient très élevé (n=200), c'est la discrétisation en 15 intervalles qui ramène le plus gros gain.

Ces résultats montrant que le choix du nombre d'intervalles à prendre pour S2 dépend fortement du nombre de données.

Si le nombre de données est faible, prendre une discrétisation trop importante fournira un gain très faible, car le système fournira des résultats biaisés, à cause du manque de données.

En revanche, si le nombre de données est suffisamment important (n=200 exemples par famille), l'utilisateur pourra se permettre de choisir une résolution plus importante. Nous voyons que le gain d'information fourni par une discrétisation en 4 intervalles va être limité à 0.85 bits, même si le nombre de données continue à augmenter. En revanche, si l'utilisateur augmente la résolution, il pourra obtenir un gain d'information plus élevé, de 1.1 bits (qui augmentera encore avec le nombre de données dans notre cas). Ainsi, en fonction du nombre de données disponibles, l'utilisateur peut choisir, en fonction du gain d'information, la discrétisation optimale de S2.

Le rejet ou la validation de H0d (qui stipule qu'une discrétisation plus importante de S2 apporte un gain plus important) dépend donc fortement du nombre de données d'entrainement dans le cas où la distribution de sortie est une loi normale. H0d peut seulement être acceptée si le nombre de données d'entrainement est atteint (environ 160 exemples par famille selon nos courbes).

Après avoir analysé les résultats obtenus en générant des données, nous allons maintenant procéder à l'analyse de l'expérience sur les données réelles.

5.4 Expérience sur les données réelles

Nous avons enfin appliqué le système sur une base de données réelle, provenant de la cohorte de l'Automne 2008 de l'École Polytechnique. La base de données originale contient 750 étudiants.

Le but de cette expérience est de tester si le système est capable de prédire la qualité de sortie du système en détectant des tendances dans les données.

Nous allons dans cette sous-partie décrire la manière dont nous avons prétraité ces données, puis comment elles ont été traitées par le système.

5.4.1 Prétraitement des données

La base de données originale ne correspond pas au schéma requis pour que le système puisse fonctionner, décrit au chapitre 3. Il nous a fallu effectuer des opérations avec les logiciels R et Excel pour adapter le format des données.

Si nous ne détaillerons toutes les opérations effectuées, deux points importants sont à soulever par rapport à ce prétraitement.

Premièrement, tous les étudiants commençant leur baccalauréat à l'Automne 2008 ne finissent pas exactement douze sessions plus tard. Certains mettent plus de temps, d'autres abandonnent :

- En ce qui concerne les étudiants ayant abandonné, aucune information ne précise l'abandon. Nous remarquons juste que les notes sont absentes à partir de la session d'abandon. Nous avons donc dû retirer ces étudiants de la base de données. Nous avons considéré comme étudiant ayant abandonné tous les étudiants ayant été absents à trois ou plus sessions d'affilée. Cela réduit la base de données de 36%. Le nombre de données traitables est donc de 480 instances.
- De plus, nous avons considéré que la dernière session du baccalauréat était la douzième session. Ainsi, les notes des étudiants mettant plus de temps à finir leur baccalauréat ont été supprimées de la base de données, ce qui peut éventuellement introduire un biais. Cependant, ce biais devrait être réduit par le fait que les notes obtenues aux sessions lors de la seconde partie du baccalauréat sont moyennées à travers la variable S2.

Deuxièmement, il y a des données manquantes dans la base de données, liées à des étudiants ayant été absents à certaines sessions, en particulier l'été. Pour pallier à ce problème, deux cas ont été considérés :

- Le premier consiste à remplacer les notes moyennes manquantes Ai par la note « 0 ». Étant donné que le système prend en compte les crédits, il devrait ainsi apprendre avec ses tables de probabilités le cas spécial correspondant à un nombre nul de crédits et une note Ai nulle.
- Le second consiste à remplacer les notes moyennes manquantes Ai par la moyenne cumulative obtenue par l'étudiant en question à la session i, de manière à ne pas « casser » la tendance. Le nombre de crédits sera quant à lui toujours nul.

Les résultats obtenus avec les deux prétraitements ont été comparés, et il se trouve que le second a fourni des résultats légèrement meilleurs. Cette méthode a donc été adoptée.

Le prétraitement est une étape importante de l'analyse de données. On considère souvent que cette étape est la plus longue dans le processus de fouille de données.

Le nombre d'exemples disponibles est réduit à 480 lors du prétraitement, ce qui est faible pour appliquer des algorithmes de reconnaissance de formes.

Si un temps important a été accordé au prétraitement des données, nous avons, dans le cadre de ce projet, dû équilibrer notre temps avec d'autres tâches, tels que la création du système et la compréhension de son fonctionnement par simulation.

Des travaux ultérieurs pourront être effectués pour optimiser davantage le prétraitement. Il faudrait également recueillir plus de données, depuis les cohortes d'autres années, de manière à obtenir de meilleurs résultats.

5.4.2 Validation croisée

Pour tester l'habilité du système à prédire les moyennes des étudiants, nous avons effectué une validation croisée, appelée « 5-fold cross validation ». Il s'agit de diviser la base de données en cinq parties. Un processus en cinq étapes est alors suivi. Il s'agit à chaque étape de séparer l'une des cinq parties des quatre autres, pour la considérer comme la base de données de test. Les quatre autres forment la base de données d'entrainement.

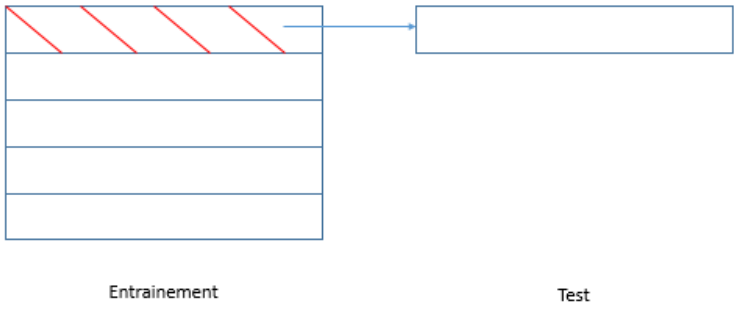


Figure 5-25 : Validation Croisée

Une fois le test effectué (l'erreur moyenne sur la base de données de test est calculée), la partie constituant les données de test est réintégrée dans la base de données d'entrainement, et une autre partie est utilisée comme base de données de test (étape suivante).

Une fois les cinq étapes effectuées, une erreur moyenne totale est calculée à partir des erreurs moyennes obtenues à chaque étape.

Nous obtenons alors l'erreur moyenne de la prédiction fournie par le système.

5.4.3 Plan d'expérience

Pour vérifier l'habilité du système à estimer la probabilité de sortie, nous allons comparer les prédictions fournies par le système avec celles obtenues par une méthode plus simple. Cette dernière correspond à la distribution de probabilités a posteriori P (S2 | Ai), soit la prédiction de S2 sachant la note obtenue à la session i. Il s'agit en fait d'un réseau bayésien possédant seulement deux variables : S2 et Ai. Nous ferons varier i, dans le but d'étudier si des sessions sont plus influentes que d'autres.

Ainsi, si la prédiction obtenue avec le réseau bayésien du système fournit une erreur plus faible que celle obtenue avec P(S2 | Ai), nous pouvons conclure que notre système apporte une valeur ajoutée en détectant les tendances des étudiants.

Ainsi, nous allons tester l'hypothèse :

H0e : La prédiction fournie par le système apporte un log-loss moyen inférieur à celui fourni par P (S2 | Ai), pour tout i de 1 à 6.

Ce test se fera après avoir optimisé les paramètres du système (discrétisation des intervalles et choix de la structure du réseau) en maximisant le gain.

5.5 Analyse des résultats obtenus avec les données réelles

Comme indiqué dans le chapitre précédent, nous procédons par validation croisée pour tester le système sur des données réelles. La première étape consiste à choisir le réseau optimal et les discrétisations optimales des variables. Nous allons ensuite nous intéresser à l'influence du département sur les données. Enfin, nous procéderons au test de l'hypothèse H0e.

Globalement, le temps de calcul pour l'apprentissage et l'inférence est de 1 seconde. Pour l'interrogation, 0.1 seconde par instance de test (CPU Quad-Core, 2.58 GB).

5.5.1 Structure de réseau et discrétisation de Ai et de S2

Suite à ce que nous avons vu sur l'évolution du gain en données simulées, nous choisissons les discrétisations de Ai et de S2 qui maximisent le gain d'information par rapport à la loi uniforme. Plusieurs valeurs ont été testées, les résultats sont synthétisés dans les tableaux ci-dessous.

Table 5-2: Gains avec un réseau bayésien naïf augmenté

	a= 5 int	a= 10 int
s2 = 2 int	0.38	0.35
s2= 3 int	0.438	0.466
s2=4 int	0.45	0.406
s2=5 int	0.494	0.426
s2=6 int	0.516	0.488
s2=7 int	0.462	0.398
s2=8 int	0.412	0.404
s2=15 int	0.344	0.346

Table 5-3: Gains avec un réseau bayésien naïf

s2 = 2 int	0.152	0.166
s2= 3 int	0.215	0.199
s2=4 int	0.286	0.27
s2=5 int	0.296	0.308
s2=6 int	0.326	0.308
s2=7 int	0.338	0.348
s2=8 int	0.3175	0.3
s2=15 int	0.258	0.156

Pour rappel, dans le cadre de ce mémoire, le gain représente la quantité d'information apportée par le système par rapport à une loi uniforme. Ainsi, plus le gain est élevé, plus le système apporte de l'information à l'utilisateur pour prédire la moyenne cumulative finale des étudiants.

La comparaison des résultats obtenus avec les deux réseaux montre que le gain est toujours plus important avec le réseau naïf augmenté qu'avec le réseau naïf. Ces résultats sont en accord avec ceux obtenus en simulation. Nous choisissons donc le réseau bayésien naïf augmenté.

En ce qui concerne la discrétisation, nous pouvons voir que le gain est optimal (0.516) avec une discrétisation de S2 en 6 intervalles, et de Ai en 5 intervalles. Le nombre de données disponibles n'est ainsi probablement pas suffisant pour avoir une résolution plus précise. Nous choisissons donc les discrétisations optimales.

5.5.2 Influence du département

Comme nous l'avons vu, le réseau bayésien naïf augmenté peut accorder une grande importance à l'influence du département lors de l'apprentissage des tables de probabilités.

Nous avons voulu tester l'importance de D, en créant un modèle naïf prenant en compte D, et un autre modèle ignorant cette variable.

Les résultats respectifs obtenus sont les suivants :

- Avec le département : le système retourne des prédictions probabilistes ayant pour log-loss moyen la valeur de 1.31.
- Sans le département : le système retourne des prédictions probabilistes ayant pour log-loss moyen la valeur de 1.22.

Ainsi, l'erreur moyenne obtenue est plus faible si l'on ignore le département que si on le prend en compte. Nous pouvons donc conclure que, d'après les données analysées, le département ne permet pas d'obtenir des prédictions plus justes de la sortie.

Deux raisons peuvent être pointées :

- Le nombre de données est restreint. De plus, les départements n'ont pas tous le même nombre de données, car le nombre d'étudiants diffère d'un département à l'autre. Ainsi, si on sépare les apprentissages, certains départements auront suffisamment de données pour permettre un apprentissage efficace, et d'autres manquerons d'exemples. Les prédictions pour les étudiants des départements en « sous-effectif » seront donc davantage biaisées.
- Peut-être que le département n'influe tout simplement pas sur la sortie.

Ainsi, nous ne prenons pas en compte le département dans le modèle, pour obtenir une erreur moyenne plus faible. La prédiction du système correspondra donc à la distribution a posteriori P(A1^C1^…^A6^C6).

5.5.3 Comparaison avec P (S2 | Ai)

Vient enfin l'étape finale, qui consiste à comparer les distributions prédictives fournies par le système avec les prédictions fournies par les probabilités conditionnelles P (S2 | Ai).

Notre système renvoi un log-loss moyen ayant une valeur de 1.22, soit un gain de 0.58, pour la probabilité a posteriori P(S2 | C1 ^ A1 ^ C2 ^ A2 ^ … ^ C6 ^ A6).

Le tableau 5.4 indique les erreurs obtenues pour différents Ai.

Table 5-4 : Log-Loss moyens obtenus sachant les moyennes de différentes sessions

	P(S2 | A1)	P(S2 | A2)	P(S2 | A3)	P(S2 | A4)	P(S2 | A5)	P(S2 | A6)
Log-Loss	1.233998	1.211569	1.392845	1.178019	1.148301	1.192245

Nous voyons que globalement, l'erreur sur la prédiction diminue avec l'avancement du processus. Cependant, des erreurs sont plus importantes pour les sessions 3 et 6. Ceci est lié au fait que ces sessions sont des sessions d'été, et possèdent donc un nombre important de données manquantes, dont le prétraitement a introduit un biais.

Nous pouvons d'autre part ajouter que le modèle P (S2 | M), avec M représentant le note moyenne obtenue lors de la première partie du baccalauréat, présente une erreur log-loss de 1.23.

Le modèle P (S2 | A5) renvoi une erreur de 1.15, soit un gain de 0.65. L'erreur renvoyée par ce modèle est donc la plus faible.

Une tentative d'interprétation peut être avancée en ce qui concerne le fait que la 5ème session est la session la plus prédictive. Cela suggère que les étudiants ont tendance à avoir des comportements qui se figent au fur et à mesure que le processus de formation avance.

D'autre part, nous observons que, avec les données réelles disponibles, le système renvoi une prédiction probabiliste de S2 *globalement* moins informative que celle fournie par la simple probabilité conditionnelle prenant en compte la note moyenne obtenue à la session 5.

Nous devons donc rejeter l'hypothèse H0e.

Au premier abord, ce résultat signifie qu'il ne semble pas y avoir de tendances dans les données autres que des tendances reconnaissables uniquement à partir de la note A5. Les variables Ai semblent toutes corrélées, dans le sens où aucune ne semble ajouter de l'information aux autres.

Nous avons tenté d'obtenir davantage de détails sur ce résultat.

Pour cela, nous avons déterminé la répartition des erreurs log-loss. En effet, les indicateurs mesurés jusqu'à présente étaient des log-loss moyennés sur l'ensemble des étudiants. Nous nous intéressons désormais aux erreurs par étudiant.

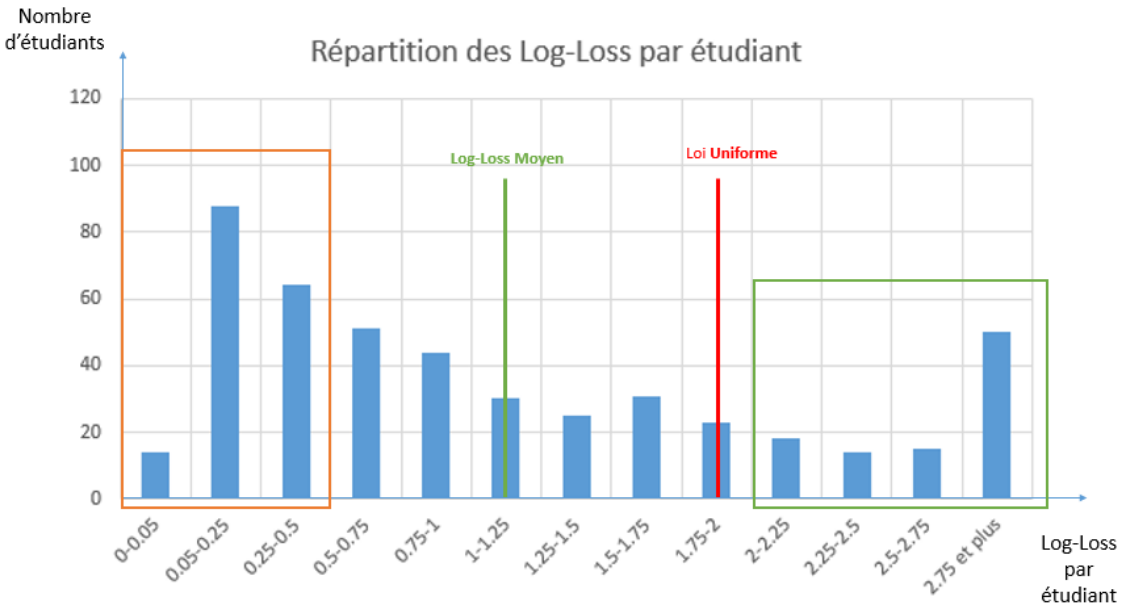


Figure 5-26 : Répartition des erreurs dans le cas des prédictions P(S2 | C1 ^ A1 ^ C2 ^ A2 ^ … ^ C6 ^ A6)

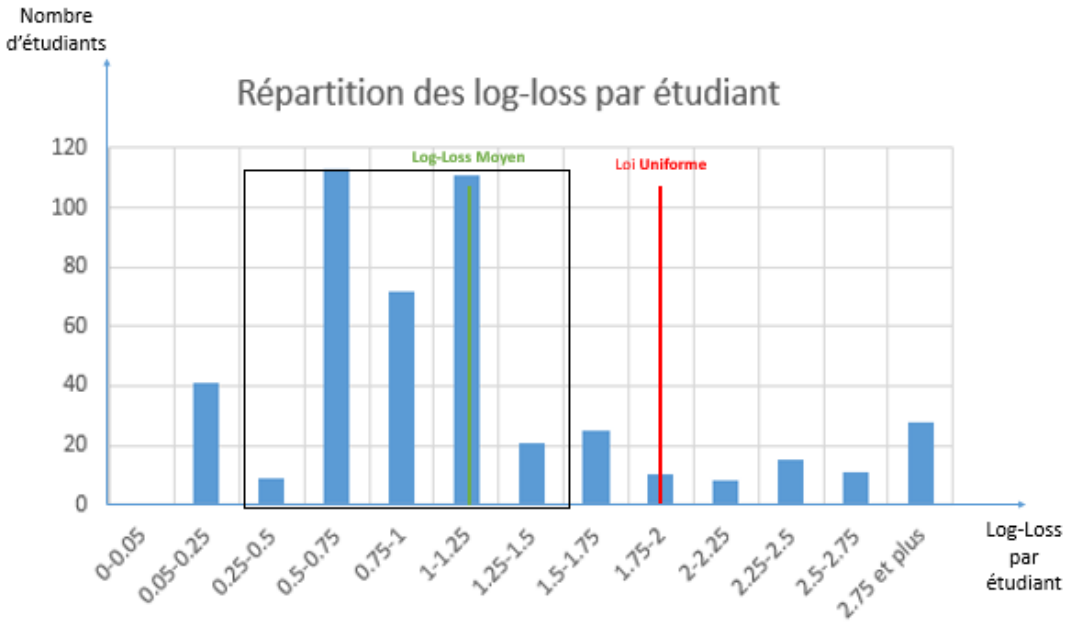


Figure 5-27 : Répartition des erreurs dans le cas de P (S2 | A5)

La ligne rouge verticale indique la valeur du log loss fournit par la loi uniforme.

Nous pouvons voir que les estimations fournies par le système sont hétérogènes, par rapport à celles fournies par P(S2 | A5). En effet, un nombre important d'étudiants (166, encadrés en orange) ont une prédiction associée très précise (log-loss proche de 0). Cependant, il y a également un nombre important (97, encadré en vert) qui ont reçu une prédiction probabiliste biaisée, ayant un log-loss plus élevé que celui de la loi uniforme.

Les prédictions de P(S2 | A5) sont plus homogènes, regroupées principalement dans l'encadré noir. Ainsi, les prédictions sont globalement moins « confiantes » que celles obtenues avec le système : il y a globalement plus d'incertitude. Mais, en contrepartie, un nombre très faible d'étudiants a reçu une prédiction ayant un log-loss plus élevé que celui de la loi uniforme, car la prédiction étant moins certaine, garde une certaine probabilité sur tous les intervalles.

Nous pouvons donc conclure que le réseau bayésien naïf augmenté a une tendance à être trop confiant dans ses prédictions. Cela est en accord avec les résultats obtenus en simulation.

D'autre part, comme nous l'avions vu lors du plan d'expérience avec les données générées, la qualité de la prédiction fournie par le système dépend fortement du nombre de données disponibles. Il est ainsi fortement possible que le nombre de données d'entrainement réduit soit l'une des causes du biais dans les résultats.

Enfin, certain des étudiants testés possèdent très probablement des comportements imprévisibles (données aberrantes), qui ne se retrouvent pas dans l'historique des données d'apprentissage. Ainsi, pour ces étudiants, l'hypothèse de fonctionnement du système n'est pas respectée, et le système ne pourra donc pas fournir de prédictions correctes.

5.6 Test en Régression

Enfin, nous avons ajouté un dernier test, basé sur un indicateur plus classique : le coefficient de détermination R^2.

Pour déterminer ce dernier, nous nous basons sur la valeur retournée par l'espérance de l'histogramme fourni par le système. Ainsi, avec la valeur réelle obtenue par l'étudiant considéré, nous pouvons obtenir le coefficient R^2.

Nous avons pour cela choisi la valeur de discrétisation de S2 correspondant au gain maximal, et de même pour les variables Ai. La valeur moyenne du coefficient R^2 obtenu en validation croisée est de 0.55 pour le réseau bayésien naïf, et de 0.58 pour le réseau bayésien naïf augmenté.

Nous avons, pour avoir un élément de comparaison, déterminé le coefficient R^2 du modèle P(S2 | M), avec M la moyenne obtenue par l'étudiant lors de la première partie du baccalauréat. La valeur du coefficient et de 0.61, ce qui suggère que le modèle basé sur la moyenne est légèrement plus explicatif que les deux autres modèles proposés, ce qui confirme que prendre séparément les notes obtenues aux différentes sessions ne fournit pas un modèle plus performant.

5.7 Conclusion de l'analyse des résultats

Les expériences ont permis de tester l'habilité du système à répondre à la problématique du mémoire, en évaluant les performances de l'outil à ressortir une estimation probabiliste de la sortie d'un processus particulier sachant les valeurs des variables observées.

Lors de l'analyse des résultats de l'expérience en données simulées, nous avons d'abord montré, en attribuant une distribution Dirac à chaque famille, que le système est capable de détecter des formes, et qu'il possède une certaine robustesse à cet égard.

En complexifiant la distribution a posteriori de la sortie (Loi normale d'écart type 0.25), nous avons montré que le système parvient à ressortir une estimation probabiliste informative de la sortie, mais que cette dernière est souvent biaisée, dans le sens où l'estimation tend à être trop confiante pour certaines valeurs.

Une comparaison des erreurs du réseau naïf avec le réseau naïf augmenté a montré que, globalement, le réseau naïf augmenté est plus performant que le réseau naïf, dans le sens où les estimations sont moins biaisées. Comme nous l'avons vu, le réseau augmenté a été construit avec l'aide de nos connaissances d'experts. Cela souligne que pour les réseaux bayésiens, l'avis d'experts importe beaucoup sur la performance. Ceci peut se présenter comme un point faible, dans le sens où des experts ne seront pas toujours présents pour l'implantation du système. D'autre part, des erreurs lors de la création du réseau peuvent mener à de graves conséquences sur la performance.

Nous avons proposé d'utiliser un indicateur que nous avons dénommé Gain, de manière à comparer les performances obtenues avec différentes discrétisations de la sortie S2. Cela nous a permis de choisir, dans le cas des données réelles, la discrétisation à prendre sur la sortie. Le choix de ce paramètre n'est pas toujours évident, et il a fallu tester plusieurs valeurs de paramètres pour obtenir le choix optimal.

L'application du système sur les données réelles a mis en évidence des limites, liées au système lui-même (comme nous l'avions déjà vu en simulation), mais aussi aux données : manque de données d'entrainement et données aberrantes. Il se trouve que le cas étudié est particulièrement ardu en ce qui concerne ces deux derniers paramètres. De nouvelles études, avec différents algorithmes (notamment des outils plus performants) pourront être effectuées sur ces données, de manière à comparer les résultats obtenus. Si les résultats ne s'améliorent pas, cela confirmera que le fait de fusionner l'information provenant des sessions passées ne permet pas d'obtenir des résultats plus informatifs qu'avec une prédiction basée seulement sur les moyennes obtenues à la session 5. Nous verrons dans la prochaine partie les perspectives de travail futur.

L'analyse des résultats obtenus avec les données réelles a également montré que les variables les plus prédictives de la moyenne S2 des étudiants sont les moyennes obtenues aux sessions 5 et 6. Ces sessions correspondent respectivement à l'Automne et l'Été de la deuxième année du baccalauréat.

CHAPITRE 6 CONCLUSION ET RECOMMANDATIONS

6.1 Synthèse des travaux de recherche

Ce projet a permis de proposer l'utilisation d'un système de prédiction probabiliste, basé sur la création et l'interrogation de réseaux bayésiens fusionnant l'information provenant de variables observées, dans le but de générer une distribution sur la variable d'intérêt. Une application sur le processus de formation des étudiants à Polytechnique Montréal a été effectuée.

La revue de littérature montre que les réseaux bayésiens sont loin de figurer en haut de la liste des méthodes les plus populaires en exploration de données industrielles. La méthode bayésienne permet au système ici proposé d'une part d'intégrer les connaissances des experts des processus avec l'apprentissage machine, pour fournir des modèles compréhensibles, et d'autre part, de fournir, de manière naturelle, des prédictions probabilistes permettant de tenir compte de l'incertitude lors de prises de décisions.

Nous avons vu que l'apport de l'expert sur le modèle influence de manière importante la qualité de la prédiction de la variable d'intérêt, en comparant les erreurs des prédictions fournies par le réseau naïf et naïf augmenté, ce dernier étant très souvent plus performant que le premier. Dans tous les cas, le système proposé dans ce mémoire a pour autre caractéristique d'être basé sur des algorithmes relativement simples, possédant seulement deux couches de variables.

La méthode suivie lors du projet a permis d'utiliser des mesures (le log loss et le gain) permettant d'évaluer le système et de choisir les paramètres maximisant son efficacité lors de son application sur un processus donné. Ainsi, la même méthode pourra être utilisée lors de l'application du système sur de nouveaux processus.

Nous avons appliqué l'outil sur le processus de formation des étudiants de Polytechnique Montréal, de manière à tester l'habilité du système à prédire, de manière probabiliste, la moyenne cumulée finale des étudiants, à partir de paramètres propres au comportement lors de la première partie du baccalauréat. L'expérience en données simulées a démontré que le système parvient bien à détecter les différents comportements, et parvient à ressortir une prédiction probabiliste informative de la sortie, si les conditions de fonctionnement sont respectées.

L'application sur les données réelles a montré que, avec les données disponibles, le système ne parvient pas à fournit une prédiction plus informative en fusionnant l'information provenant de plusieurs sessions qu'en prenant seulement en compte la moyenne obtenue à la 5ème session.

D'autre part, nous avons vu que cette dernière moyenne est l'information la plus influente sur la sortie du processus. En effet, les prédictions obtenues en prenant en compte les sessions précédentes présentaient des erreurs plus importantes. Cela suggère que les comportements des étudiants ont tendance à se figer au fur et à mesure que le processus avance.

6.2 Limites du système et perspectives d'amélioration

Le système proposé se trouve cependant confronté à certaines limites.

Tout d'abord l'expérience en données simulées montre que, même si le système parvient parfaitement à détecter les tendances, l'estimation probabiliste de la sortie peut être biaisée, dans le sens où l'algorithme a tendance à être trop confiant, et à négliger les probabilités les moins élevées. Ceci peut se révéler problématique, dans le sens où certains risques éventuels de baisse de qualité peuvent apparaitre comme non-existants sur la prédiction du système, alors qu'ils sont bel et bien présents. Comme nous l'avons vu, fournir une estimation non biaisée de la sortie n'est pas une tâche reconnue comme simple dans la littérature. Plusieurs travaux de recherche en apprentissage machine suggèrent l'utilisation de méthodes dites de calibration de l'estimation probabiliste, en utilisant des approches telles que la régression isotonique (Niculescu-Mizil & Caruana , 2005). Ces travaux mettent également en avant l'utilisation de méthodes d'apprentissage machine qui diffèrent des méthodes probabilistes, en particulier les machines à vecteurs de support (SVM). Si ces méthodes ne sont pas « naturellement probabilistes », contrairement aux réseaux bayésiens, il est possible de leur faire retourner des prédictions probabilistes, avec l'utilisation d'approches comme « Platt Scaling » (Platt, 1999). Les résultats obtenus ont montré des estimations particulièrement justes (les meilleurs résultats sont ceux obtenus par les SVM). Il serait ainsi intéressant, lors de recherches futures, d'appliquer des algorithmes basés sur ces méthodes pour comparer les résultats. La librairie Python Scikit-Learn permet, entre autres, l'implémentation de ces approches. Cependant, toutes ces méthodes de calibrations requièrent généralement un nombre important de données pour assurer un fonctionnement efficace. D'autre part,

l'implémentation devient plus complexe, et l'effet « boite noire » réapparait, dans le sens ou les modèles fournis pourront sembler complexes à comprendre. Un choix devra donc être fait, en ce qui concerne le compromis entre l'intelligibilité du modèle et la performance des résultats.

Ensuite, une limite a été mise en évidence lors de l'application du système sur les données réelles. En effet, les prédictions obtenues avec le système n'étaient pas plus informatives qu'une simple prédiction probabiliste obtenue à partir d'une seule variable observée. Comme nous l'avons vu, l'une des causes pourrait être liée à l'absence de formes (autres que des formes constantes) dans les données. Cependant, un biais dans les prédictions du système a été mis en évidence (prédictions trop confiantes). Ce biais peut être lié aux limites de l'algorithme, comme ci-dessus, mais aussi au manque de données d'entrainement, étant donné que ce paramètre influence de manière importante la qualité des prédictions. Ainsi, il serait intéressant de réunir davantage de données, provenant de cohortes antérieures, pour observer si une amélioration dans les prédictions survient.

Il serait d'autre part intéressant d'ajouter davantage d'informations en entrée. Les études effectuées dans la littérature en EDM utilisaient souvent des informations relatives aux conditions sociales des étudiants, comme le pays d'origine, ou le statut de résident, qui se révélaient influentes sur la sortie. Nous pourrons également rechercher des informations relatives aux cours pris aux différentes sessions.

Ensuite, une application du système sur de nouveaux processus, dans le domaine manufacturier ou des services, serait très intéressante. Il doit s'agir, comme nous l'avons précisé dès l'introduction, de processus ayant un nombre très élevé d'exemples par rapport au nombre de variables. Les industries actuelles tendent graduellement à s'orienter vers des industries dites 4.0, où de nombreux objets connectés apporteront de l'information sur les processus en cours. Ce terrain devrait constituer une véritable mine d'or pour l'application de systèmes tels que celui proposé dans ce mémoire.

Pour conclure, ce travail peut être vu comme un premier pas vers un système présentant une prédiction probabiliste de la qualité de sortie à l'utilisateur, de manière à visualiser les différentes possibilités d'évolution de la production en cours avec les estimations associées. Cela permettrait d'agir avec, plutôt que à la place de, ce dernier pour améliorer la qualité de production, en particulier dans des situations où l'incertitude est importante, et où les décisions ne sauraient être prises hâtivement.

BIBLIOGRAPHIE

al, B. B. (2014). Joint production and quality control of unreliable batch manufacturing systems with rectifying inspection. *International Journal of Production Research*, 4103-4117.

Ali, O., & Chen, Y. (1999). Design quality and robustness with neural networks. *IEEE Transactions on Neural Networks, 10(6)*, 1518–1527.

Bassetto, S., Paredes, C., & Baud-Lavigne, B. (2013). A systemic approach of quality controls. *7th Annual IEEE Systems Conference*.

Baud-Lavigne, B., Bassetto, S., & Agard, B. (2014). A method for a robust optimization of joint product and supply chain design. *Journal of Intelligent Manufacturing*.

Bekele, R., & Menzel, W. (2005). A Bayesian Approach to Predict Performance of a Student (BAPPS): A Case with Ethiopian Students. *IASTED International Conference on Artificial Intelligence and Applications*.

Besheti, B., & Desmarais, M. C. (2015). Goodness of fit of skills assessment approaches:Insights from patterns of real vs. synthetic data sets. *INternational Educational Data Mining*, 81-86.

Bessière, P., Mazer, E., Ahuactzin, J. M., & Mekhnacha, K. (2013). *Bayesian Programming.* CRC press.

Bettayeb, B., Bassetto, S., Vialletelle, P., & Tollenaere, M. (2012). Quality and exposure control in semiconductor manufacturing. Part I: Modelling. *International Journal of Production Research*, 6835-6851.

Bettayeb, B., Bassetto, S., Vialletelle, P., & Tollenaere, M. (2012). Quality and exposure control in semiconductor manufacturing. Part II: Evaluation. *International Journal of Production Research*, 6852-6869.

Bishop, C. (2006). *Pattern Recognition and Machine Learning.* Springer.

Bouaziz, M. F., Zamai, E., & Duvivier, F. (2013). Towards Bayesian Network Methodology for Predicting the Equipment Health Factor of Complex Semiconductor Systems. *International Journal of Production Research*, 4597-4617.

Bouslah, B., Ghrabi, A., & Pellerin, R. (2014). Joint production and quality control of unreliable batch manufacturing systems with rectifying inspection. *International Journal of Production Research*, 4103-4117.

Cheng, J., & Greiner, R. (1999). Coparing Bayesian Network Classifiers. *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, 101-108.

Delen, D. (2010). A comparative analysis of machine learning techniques for student retention management. *Decision Support Systems*, 498–506.

Desmarais, M. C., Naceur, R., & Behsheti, B. (2012). Linear Models of student skills for static data. *User Modelling, Adaptation and Personalization*.

Domingos, P., & Pazzani, M. (1996). Beyons Independance: Conditions for the optimality of the simple Bayesian Classifier. *Proceedings ofthe Fifth International Conference on Knowledge Discovery and Data Mining*.

Domingos, P., & Pazzani, M. (1997). On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. *Machine Learning, 29*, 103-130.

Eisenstein, E., & Alemi, F. (1996). A comparison of three techniques for rapid model development: an application in patient risk-stratification. *Proc/AMIA Annu Fall Symp*, 443–447.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth , P. (1996). From Data Mining to Knowledge Discovery in Databases. *AI MAGAZINE*, 37-54.

Fenton, N., Neil, M., & Marquez, D. (2008). Using Bayesian networks to predict software defects and reliability. *Journal of Risk and Reliability*, 701-712.

Friedman , N., & Goldszmidt, M. (1996). Building Classifiers using Bayesian Networks . *Proceeding of the National Conference on AI*, 1277-1284.

Grove, W. A., & Wasserman, T. (2004). The Life Cycle Pattern of Collegiate GPA: Longitudinal Cohort Analysis and Grade Inflation. *Journal of Economic Education*, 162-174.

Haddawy, P., Thi, N., & Hien, T. N. (2007). A decision support system for evaluating international student applications. *Proc. Frontiers Educ. Conf.*, 1-4.

Harding, J. A., Shahbaz, M., Srivinas, & Kusiak, A. (2006). Data Mining in Manufacturing:A Review. *Journal of Manufacturing Science and Engineering*, 969-976.

Heckermann, D., Geiger, D., & Chickering, D. M. (1995). Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Microsoft Research MSR-TR-94-09*.

Khan, A. A., & Moyne, J. R. (2007). An Approach for Factory-Wide Control Utilizing Virtual Metrology. *IEEE Transactions on Semiconductor Manufacturing*, 364-374.

Khan, A. A., Moyne, J. R., & Tilbury, D. M. (2007). An Approach for Factory-Wide Control Utilizing Virtual Metrology. *IEEE Transactions on Semiconductor Manufacturing Vol. 20*, 364 - 375.

Koskal, G., Batmaz, I., & Tetik, M. C. (2011). A review of data mining applications for quality improvement. *Expert Systems with Applications*, 13448–13467.

Lebeltel, O., Bessière, P., Diard, J., & Mazer, E. (2004). Bayesian Robot Programming. *Autonomous Robots, Springer Verlag*, 49–79.

Lee, S.-M., & Abbott, P. A. (2003). Bayesian networks for knowledge discovery in large datasets: basics for nurse researchers. *Journal of Biomedical Informatics*, 389–399.

Lucas, P., Van Der Gaag, L., & Hanna, A. (2004). Bayesian networks in biomedicine and healthcare. *Artificial Intelligence in Medicine*, 201–214.

Morgan, M. G., & Henrion, M. (1990). *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis.* Cambridge: Cambridge University Press.

Niculescu-Mizil, A., & Caruana , R. (2005). Predicting Good Probabilities With Supervised Learning. *ICML Proceedings of the 22 nd International Conference on Machine Learning*, 625 - 632 .

Orr, M. K. (2011). Performance trajectory of students in the engineering disciplines. *Frontiers in Education Conference (FIE)*, S3H-1 - S3H-4.

Perzyk, M., Biernacki, R., & Kochanski, A. (2005). Modeling ofmanufacturing processesby learning systems: The naive bayesian classifier versus artificial neural networks. *Journal of Materials Processing Technology*, 164–165.

Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 61–74.

Restrepo-Moreno, D., Charron-Latour, J., Pourmonet, H., & Bassetto, S. (Accepted 2015). Seizing opportunities for change at the operational level. *International Journal of Health Care Quality Assurance*.

Romero, C., & Ventura, S. (2010). Educational Data Mining: A Review. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, 601-617.

Sahnoun, M., Bettayeb, B., & Bassetto, S. (2014). Simulation-based optimization of sampling plans to reduce inspections while mastering the risk exposure in semiconductor manufacturing. *Journal of Intelligent Manufacturing*, 1-15.

Sharabiani, A. (2014). An enhanced bayesian network model for prediction of students' academic performance in engineering programs. *IEEE Global Engineering Education Conference (EDUCON)*, 832 - 837.

Tilouche , S., Bassetto, S., & Partovi-Nia, V. (2014). Classification Algorithms for Virtual Metrology. *IEEE Management of Innovation and Technology*, 495 - 499.

Tiwari, A., Turner, C., & Majeed, B. (2008). A review of business process mining: state-of-the-art and future trends. *Business Process Management Journal vol. 14*, 5 - 22.

Tosun, A., Bener, A. B., & Akbarinasaji, S. (2015). A systematic literature review on the applications of Bayesian networks to predict software quality. *Software Quality Journal*, 1-33.

Tsai, Y. H., Chen, J., & Lou, S. (1999). An in-process surface recognition system based on neural networks in end milling cutting operations. *International Journal of Machine Tools and Manufacture, 39(4)*, 583–605.

Weiss, S. M., Dhurandhar, A., & Baseman, R. J. (2013). *Improving Quality Control by Early Prediction of.* Yorktown Heights: IBM.

ANNEXES

ANNEXE A – DÉTAILS SUR LA GÉNÉRATION DE DONNÉES

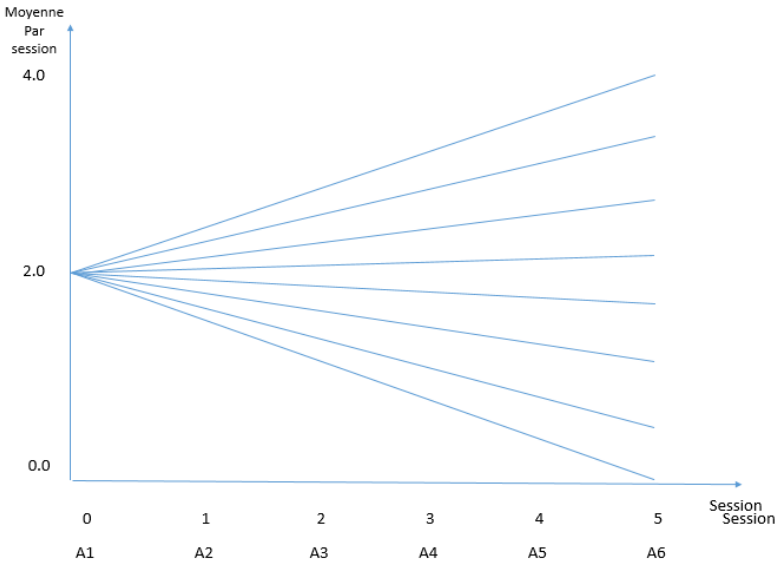


Figure i : Familles type I

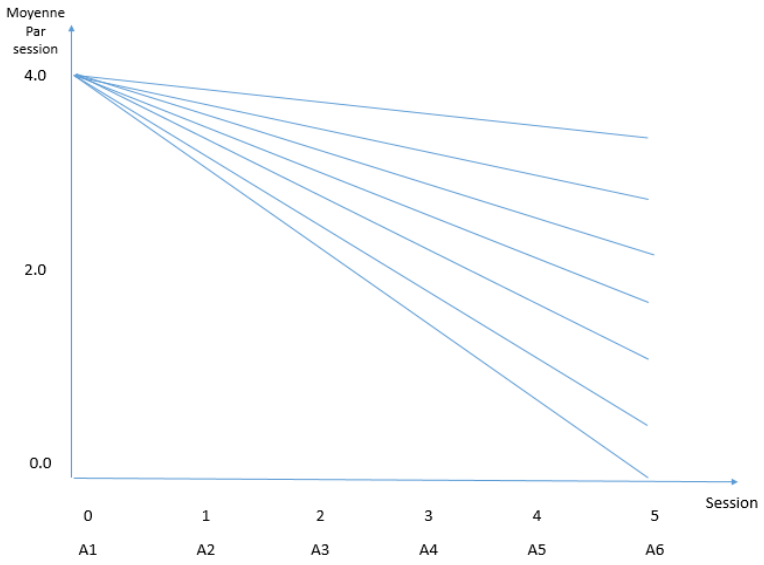


Figure ii : Familles type II

Nous générons les familles de tendances à partir des courbes représentées sur les figures i et ii. Chaque courbe correspond à une famille particulière.

Chaque famille de tendances « a » est générée séparément. Pour chaque famille, on génère des données pour chaque variable Ai, sous forme de vecteur de taille m, selon l'équation suivante :

$$\mathbf{Ai} = \mathbf{Ai\text{-}1} + f(a, i, \text{origine})\,\mathbf{U} - 0.03\,\mathbf{Ci} + b\,\mathbf{U}$$

Les valeurs du vecteur Ci sont prises uniformément dans [1,6] et [12, 18], respectivement s'il s'agit d'une session d'été, ou d'Automne/Hiver.

La fonction f(a) a pour paramètres « a » et « origine ». Ainsi, il y a 16 fonctions f(a) différentes, une par courbe, les courbes étant représentées sur les figures i et ii.

Les équations des courbes sont les suivantes :

a) Courbes de type I

F(a, i, origine=2) = a*i + 2, avec a prenant pour valeurs {-0.4;-0.3;-0.2;-0.1;0.1;0.2;0.3;0.4}

b) Courbes de type II

F(a, i, origine=4) = a*i + 4, avec a prenant pour valeurs {-0.8;-0.7;-0.6;-0.5;-0.4;-0.3;-0.2;-0.1}

A chaque famille « a » on associe (séparément) un vecteur S2, aussi de taille m. Ce vecteur est généré de la manière suivante :

$$\mathbf{S2}\,(a) = \text{Distrib}\,(a) \times \mathbf{U}$$

Où Distrib correspond à une distribution de probabilités propre à la famille « a », qui simule l'incertitude. Au cours du plan d'expérience, nous considérerons 3 cas :

a) Distrib (a) = Dirac (h) : La distribution est un Dirac, centré sur la valeur h. La valeur de h, propre à a.

b) Distrib (a) = Normale(h, 0.25) : La distribution est une loi Normale, centrée sur la valeur h, d'écart type 0.25, pour simuler de l'incertitude.

c) Distrib (a) = Uniforme ([0,4]) : La distribution est une loi Uniforme entre 0 et 4. Il s'agit d'un cas d'incertitude critique, c'est-à-dire que la connaissance de la tendance d'entrée n'explique en rien la variation en sortie.

La valeur du paramètre h est distribuée de la manière suivante.

Pour les familles de la figure i (type I), la paramètre h de chaque famille « a » est égal à la valeur h = F(a, 5, origine=2) = a*5 + 2. Ainsi, chaque famille de type I a son paramètre h associé, et donc sa distribution de sortie centrée autour d'une valeur qui lui est associée.

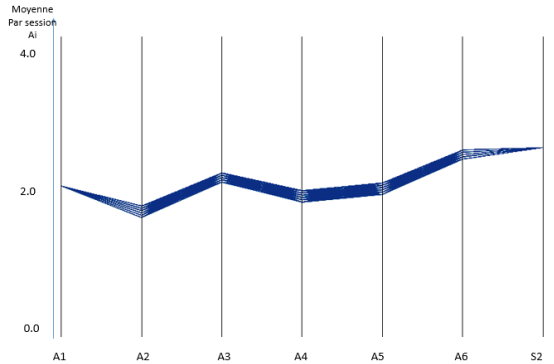
Pour les familles de la figure ii (type II), le paramètre h de chaque famille « a » est égal à la valeur h = -a*5. Nous avons choisi que le centre de la distribution de sortie associée ne soit pas égal à F(a,5, ordonnée=4) = a*5 + 4.

La raison est que nous souhaitons limiter le fait que la distribution d'une famille de Type I ayant pour tendance d'entrée F(a, 5, origine=2) = a*5 + 2 et que la distribution d'une famille de Type II ayant pour tendance d'entrée F(a_autre, 5, origine=4) = a_autre*5 + 4 telles que F(a, 5, origine=2)= F(a_autre, 5, origine=4) aient le même paramètre h (c'est-à-dire soient centrées autour de la même valeur), de manière à ce que A6 ne soit pas le seule variable prédictive de la sortie.

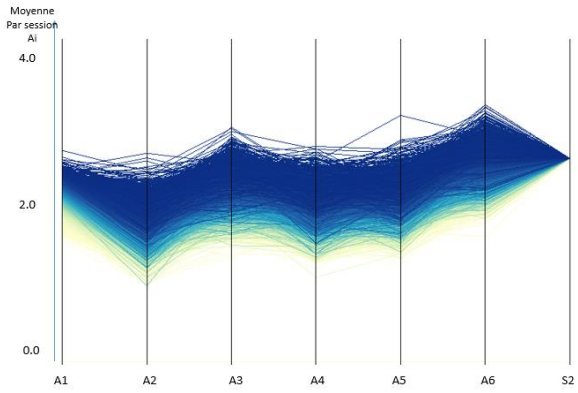
Les figures ci-dessous présentent une illustration de la génération d'une famille de tendance. Une seule famille de tendances est ici présentée (avec 200 exemples), ayant une courbe ayant pour fonction F(0.1, i, origine=2) = 0.1*i + 2. Nous faisons varier les paramètres Sigma_Bruit, et la distribution de sortie.

Ainsi, le lecteur doit bien savoir que la base de données d'apprentissage générée contient 16 familles similaires, basées sur les courbes décrites ci-dessus.

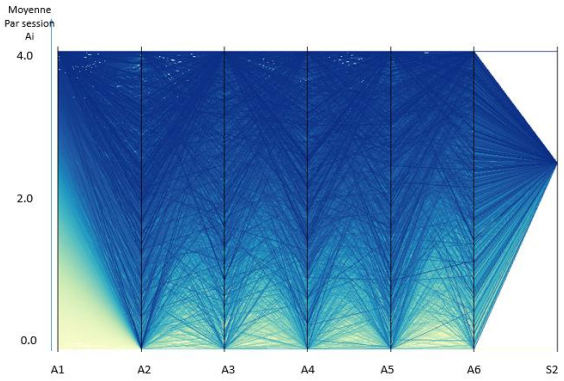
Les distorsions initiales sont liées à l'influence des crédits.



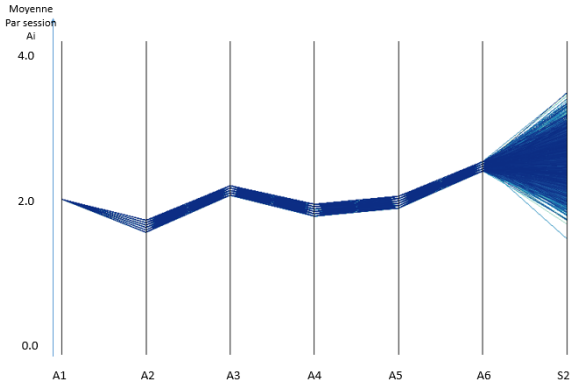
Tendance a: Sigma_Bruit = 0, Distribution_Sortie = Dirac (n)



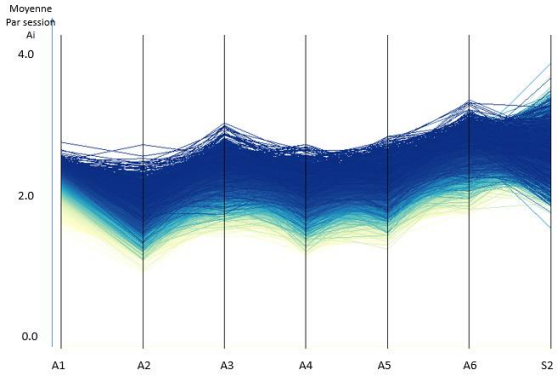
Tendance a: Sigma_Bruit = 0.15, Distribution_Sortie = Dirac (n)



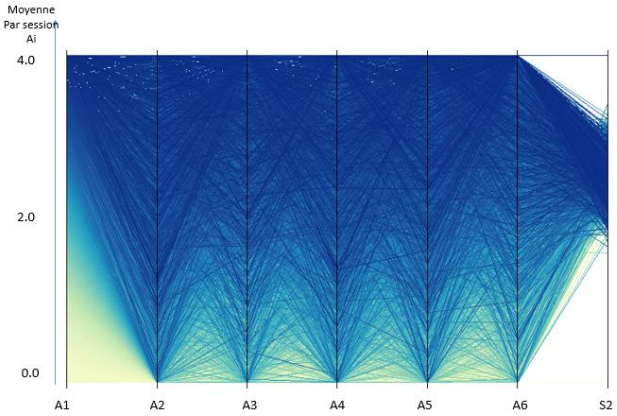
Tendance a: Sigma_Bruit = 1.0, Distribution_Sortie = Dirac (n)



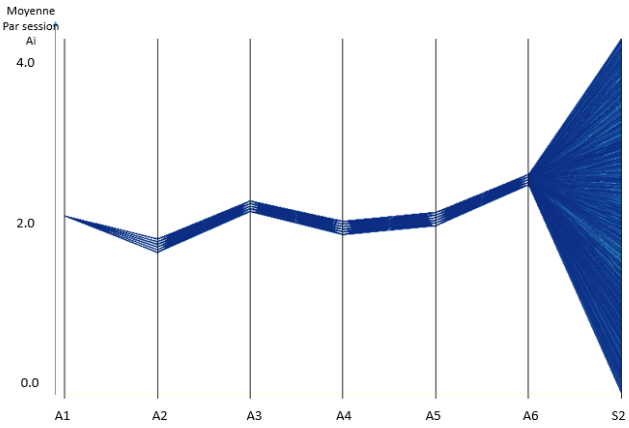
Tendance a: Sigma_Bruit = 0, Distribution_Sortie = Loi_Normale (n, 0.25)



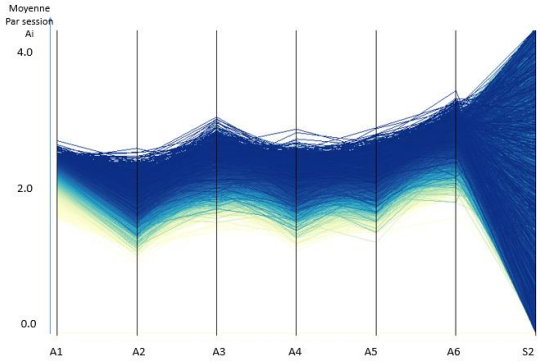
Tendance a: Sigma_Bruit = 0.15, Distribution_Sortie = Loi_Normale (n, 0.25)



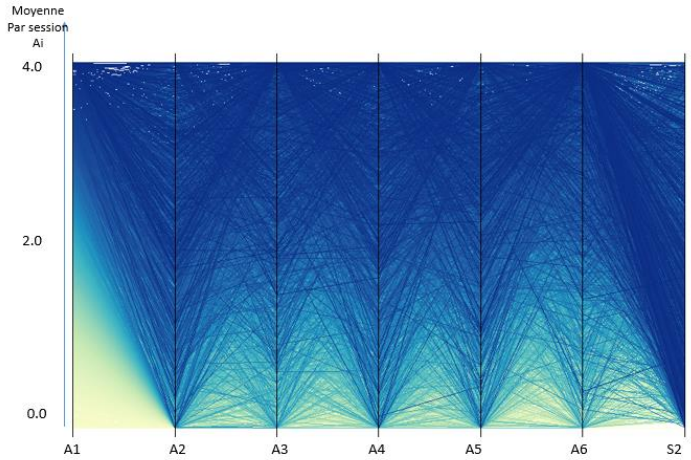
Tendance a: Sigma_Bruit = 1.0, Distribution_Sortie = Loi_Normale (n, 0.25)



Tendance a: Sigma_Bruit =0, Distribution_Sortie = Loi_Uniforme(0,4)



Tendance a: Sigma_Bruit =0.15, Distribution_Sortie = Loi_Uniforme(0,4)



Tendance a: Sigma_Bruit =1.0, Distribution_Sortie = Loi_Uniforme(0,4)

ANNEXE B – RESULTATS OBTENUS EN SIMULATION

Si la distribution a posteriori est un Dirac :

Sigma Bruit	Nombre d'Exemples	Nombre de divisions de S2	Reseau Naif Augmente			Reseau Naif			Log Loss Loi Uniforme
			Avg LL	Gain		Avg LL	Gain		
0	5	4	0.245812	1.104188		0.11363	1.23637		1.35
	20		0.0415239	1.308476		0.02916	1.32084		1.35
	200		0.0065673	1.343433		0.01857	1.33143		1.35
	5	8	0.4904407	1.579559		0.303	1.767		2.07
	20		0.061702	2.008298		0.0536	2.0164		2.07
	200		0.0031611	2.066839		0.00951	2.06049		2.07
	5	15	0.5407883	2.159212		0.31445	2.38555		2.7
	20		0.0635043	2.636496		0.058	2.642		2.7
	200		0.0034555	2.696544		0.01096	2.68904		2.7
0.15	5	4	0.7844068	0.565593		0.51494	0.83506		1.35
	20		0.4874923	0.862508		0.39509	0.95491		1.35
	200		0.2916269	1.058373		0.36036	0.98964		1.35
	5	8	1.4096681	0.660332		1.05644	1.01356		2.07
	20		0.9017831	1.168217		0.65125	1.41875		2.07
	200		0.5845175	1.485483		0.63692	1.43308		2.07
	5	15	1.5494155	1.150584		1.10992	1.59008		2.7
	20		0.9923778	1.707622		0.72485	1.97515		2.7
	200		0.6392818	2.060718		0.75078	1.94922		2.7
1	5	4	1.4342867	-0.08429		1.81263	-0.46263		1.35
	20		1.5729054	-0.22291		1.76843	-0.41843		1.35
	200		1.4420951	-0.0921		1.55	-0.2		1.35
	5	8	2.1034944	-0.03349		2.6269	-0.5569		2.07
	20		2.2573119	-0.18731		2.58493	-0.51493		2.07
	200		2.2084517	-0.13845		2.36	-0.29		2.07
	5	15	2.5993494	0.100651		3.25	-0.55		2.7
	20		2.7564264	-0.05643		3	-0.3		2.7
	200		2.6183036	0.081696		2.8	-0.1		2.7

Si la distribution a posteriori est une loi normale d'écart-type 0.25 :

Sigma Bruit	Nombre d'Exemples	Nombre de divisions de S2	Reseau Naif Augmente			Reseau Naif			Log Loss Loi Uniforme	Log Loss Distribution Réelle	Gain Distribution Réelle
			Avg LL	Gain		Avg LL	Gain				
0	5	4	0.72888886	0.6211111		0.76485	0.58515		1.35	0.28	1.07
	20		0.40064425	0.9493557		0.54827	0.80173		1.35	0.28	1.07
	200		0.35269015	0.9973098		0.47973	0.87027		1.35	0.28	1.07
	5	8	1.37574893	0.6942511		1.55955	0.51045		2.07	0.72	1.35
	20		1.04473249	1.0252675		1.11593	0.95407		2.07	0.72	1.35
	200		0.85177369	1.2182263		0.95309	1.11691		2.07	0.72	1.35
	5	15	2.38665949	0.3133405		3.35961	-0.6596		2.7	1.3	1.4
	20		1.88645733	0.8135427		2.15476	0.54524		2.7	1.3	1.4
	200		1.48217638	1.2178236		1.57072	1.12928		2.7	1.3	1.4
0.15	5	4	0.9741686	0.3758314		0.7546	0.5954		1.35	0.28	1.07
	20		0.60965239	0.7403476		0.68902	0.66098		1.35	0.28	1.07
	200		0.47147346	0.8785265		0.65175	0.69825		1.35	0.28	1.07
	5	8	1.77174102	0.298259		1.67967	0.39033		2.07	0.72	1.35
	20		1.27205331	0.7979467		1.40232	0.66768		2.07	0.72	1.35
	200		1.01387755	1.0561224		1.24114	0.82886		2.07	0.72	1.35
	5	15	2.55995207	0.1400479		3.01858	-0.3186		2.7	1.3	1.4
	20		2.16213489	0.5378651		2.17023	0.52977		2.7	1.3	1.4
	200		1.5841731	1.1158269		1.7582	0.9418		2.7	1.3	1.4
1	5	4	1.78	-0.43		1.4404	-0.0904		1.35	0.28	1.07
	20		1.59	-0.24		1.26986	0.08014		1.35	0.28	1.07
	200		1.2	0.15		1.34342	0.00658		1.35	0.28	1.07
	5	8	2.56	-0.49		2.67647	-0.6065		2.07	0.72	1.35
	20		2.27	-0.2		2.38874	-0.3187		2.07	0.72	1.35
	200		2.12	-0.05		2.16225	-0.0922		2.07	0.72	1.35
	5	15	3.056	-0.356		3.7556	-1.0556		2.7	1.3	1.4
	20		2.82879244	-0.128792		3.2829	-0.5829		2.7	1.3	1.4
	200		2.75	-0.05		2.92301	-0.223		2.7	1.3	1.4

Si la distribution a posteriori est une loi uniforme :

Sigma Bruit	Nombre d'Exemples	Nombre de divisions de S2	Reseau Naif Augmente			Reseau Naif			Log Loss Loi Uniforme
			Avg LL	Gain		Avg LL	Gain		
0 Noise	5	4	1.460869	-0.111		1.8044	-0.454		1.35
	20		1.424491	-0.074		1.5261	-0.176		1.35
	200		1.370186	-0.02		1.3982	-0.048		1.35
	5	8	2.202954	-0.133		2.7647	-0.695		2.07
	20		2.174679	-0.105		2.3916	-0.322		2.07
	200		2.103478	-0.033		2.1079	-0.038		2.07
	5	15	2.863735	-0.164		3.7132	-1.013		2.7
	20		2.821668	-0.122		3.1752	-0.475		2.7
	200		2.755414	-0.055		2.7775	-0.077		2.7
0.15 Noise	5	4	1.498881	-0.149		1.8696	-0.52		1.35
	20		1.510171	-0.16		1.5844	-0.234		1.35
	200		1.415184	-0.065		1.4331	-0.083		1.35
	5	8	2.198895	-0.129		2.7667	-0.697		2.07
	20		2.250464	-0.18		2.4817	-0.412		2.07
	200		2.156994	-0.087		2.1832	-0.113		2.07
	5	15	2.801048	-0.101		3.3605	-0.661		2.7
	20		2.91906	-0.219		3.0819	-0.382		2.7
	200		2.786218	-0.086		2.796	-0.096		2.7
1.0 Noise	5	4	1.570408	-0.22		1.8431	-0.493		1.35
	20		1.787692	-0.438		1.5902	-0.24		1.35
	200		1.36	-0.01		1.4042	-0.054		1.35
	5	8	2.239476	-0.169		2.8458	-0.776		2.07
	20		2.507039	-0.437		2.5541	-0.484		2.07
	200		2.116781	-0.047		2.1271	-0.057		2.07
	5	15	2.880735	-0.181		3.7887	-1.089		2.7
	20		2.997765	-0.298		3.4132	-0.713		2.7
	200		2.818047	-0.118		2.8039	-0.104		2.7